Proceedings

2020 6th International Symposium on System and Software Reliability

ISSSR 2020

Proceedings

2020 6th International Symposium on System and Software Reliability

ISSSR 2020

Chengdu, China
24-25 October 2020

IEEE COMPUTER SOCIETY CONFERENCE PUBLISHING SERVICES

Los Alamitos, California
Washington • Tokyo

IEEE COMPUTER SOCIETY

ISBN-13: 978-0-7381-0497-3
BMS Part# CFP20J16-ART

Additional copies may be ordered from:

IEEE Computer Society
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314
Tel: + 1 800 272 6657
Fax: + 1 714 821 4641
http://computer.org/cps
cps@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: + 1 732 981 0060
Fax: + 1 732 981 9667
http://shop.ieee.org/store/
customer-service@ieee.org

IEEE Computer Society
Asia/Pacific Office
Watanabe Bldg., 1-4-2
Minami-Aoyama
Minato-ku, Tokyo 107-0062
JAPAN
Tel: + 81 3 3408 3118
Fax: + 81 3 3408 3553
tokyo.ofc@computer.org

Editorial production by Cristina Ceballos
Cover art production by Hector Torres

IEEE Computer Society
***Conference Publishing Services* (CPS)**
http://www.computer.org/cps

2020 6th International Symposium on System and Software Reliability (ISSSR)

ISSSR 2020

Table of Contents

Session I: AI for System Analysis and Evaluation

Session II: Algorithms, Models, and Techniques for System Construction

Session III: System and Application

Session IV: Performance, Trustworthiness, and Availability of System Design

Session V: Reliability, Security, and Quality

Message from the Steering Committee Chair, General Chairs, and Program Chairs
ISSSR 2020

The COVID-19 pandemic has made a significant impact on many things. ISSSR 2020, the Sixth IEEE International Symposium on System and Software Reliability, is not an exception. Instead of having an in-person meeting in Chengdu, China as originally planned, we will have an online conference on October 24 and 25 using Zoom.

Although attendees can only have virtual interactions with each other via Internet, the conference has taken extra steps to not only organize sessions for paper presentations and discussions, but also create special chat rooms on selected topics for interested participants to exchange their experiences and lessons learned from research and practical work, as well as to review and explore the best and most efficient techniques for the development of reliable, secure, and trustworthy systems.

In addition to 26 papers in the conference proceedings published by IEEE, seven papers are recommended to the *International Journal of Performability Engineering* (IJPE), an Ei-indexed journal. These papers cover a broad spectrum from security, safety, reliability, to quality assurance of systems and software. There are also two keynote speeches:

- *Faults, Failures, and Vulnerabilities: What are the Trends and How Do We Make Progress?* by Dr. D. Richard Kuhn from US National Institute of Standards and Technology

- *Human-Machine Pair Programming: An Intelligent and Automated Approach for Software Productivity and Reliability* by Professor Shaoying Liu from Hiroshima University, Japan

ISSSR 2020 is technically sponsored by the IEEE Reliability Society and organized by the University of Electronic Science and Technology of China.

We would like to thank the organizing and the local committees for managing all the logistics, program committee for evaluating papers and accepting those of high quality, and more importantly all the attendees for their participation and support. We hope that you will benefit from the technical papers and their presentations. We also hope that you enjoy the online discussions with your colleagues.

W. Eric Wong, *University of Texas at Dallas, USA*
ISSSR 2020 Steering Committee Chair

Yuanshun Dai, *University of Electronic Science and Technology of China, China*
Qiang Miao, *Sichuan University, China*
ISSSR 2020 General Chairs

Fevzi Belli, *University of Paderborn, Germany*
Liang Luo, *University of Electronic Science and Technology of China, China*
ISSSR 2020 Program Chairs

Organizing Committee
ISSSR 2020

General Chairs
Yuanshun Dai, *University of Electronic Science and Technology of China, China*
Qiang Miao, *Sichuan University, China*

Program Chairs
Fevzi Belli, *University of Paderborn, Germany*
Liang Luo, *University of Electronic Science and Technology of China, China*

Publicity Chairs
Michael Grottke, *Friedrich-Alexander University, Germany*
Peng Sun, *University of Electronic Science and Technology of China, China*

Local Chair
Xiwei Qiu, *University of Electronic Science and Technology of China, China*

Publication Chair
Chuan Li, *Chongqing Technology and Business University, China*

Finance Chairs
W. Eric Wong, *University of Texas at Dallas, USA*
Siwei Zhou, *Wuhan University of Technology, China*

Secretariat
Dongcheng Li, *University of Texas at Dallas, USA*
Linghuan Hu, *University of Texas at Dallas, USA*

Web Master
Shou-Yu Lee, *University of Texas at Dallas, USA*

Program Committee
ISSSR 2020

Jun Ai, *Beihang University, China*
Doo-Hwan Bae, *Korea Advanced Institute of Science and Technology, Korea*
Mark Bentsen, *Argo Data, USA*
Lon Chase, *IEEE Reliability Society, USA*
Yixiang Chen, *East China Normal University, China*
Zhenyu Chen, N*anjing University, China*
Byoungju Choi, *Ewha Womans University, Korea*
William Chu, *Tunghai University, Taiwan*
Sunita Chulani, *Cisco, USA*
Vidroha Debroy, *AT&T (USA), USA*
Junhua Ding, *University of North Texas, USA*
Tadashi Dohi, *Hiroshima University, Japan*
Jian Dong, *Harbin Institute of Technology, China*
Wei Dong, *National University of Defense Technology, China*
Yunwei Dong, *Northwestern Polytechnical University, China*
Lance Fiondella, *University of Massachusetts Dartmouth, USA*
Ruizhi Gao, *Sonos Inc., USA*
Bing Guo, *Sichuan University, China*
Tom Hill, *The Fellows Consulting Group, USA*
Birgit Hofer, *Graz University of Technology, Austria*
Chin-Yu Huang, *National Tsing Hua University, Taiwan*
Zhao Ji, *Guangdong Ocean University, China*
Chuan Li, *Chongqing Technology and Business University, China*
Jenny Li, *Kean University, USA*
Steve Li, *Western New England University, USA*
Yihao Li, *Graz University of Technology, Austria*
Yun Lin, *Harbin Engineering University, China*
Shaoying Liu, *Hiroshima University, Japan*
José Maldonado, *University of São Paulo, Brazil*
Nick Multari, *Pacific Northwest National Laboratory, USA*
Manuel Nuñez, *Universidad Complutense de Madrid, Spain*
Pete Rotella, *Cisco, USA*
Mike Siok, *University of Texas at Arlington, USA*
Hongwei Tao, *Zhengzhou University of Light Industry, China*
Nguyen Tien, *University of Texas at Dallas, USA*
Tugkan Tuglular, *Izmir Institute of Technology, Turkey*
Auri Vincenzi, *Federal University of São Carlos, Brazil*
Jian Wang, *Chinese Academy of Sciences, China*
Yong Wang, *Anhui University of Engineering, China*
Ziyuan Wang, *Nanjing University of Posts and Telecommunications, China*
Franz Wotawa, *Graz University of Technology, Austria*
Qinggang Wu, *Zhengzhou University of Light Industry, China*
Jianwen Xiang, *Wuhan University of Technology, China*
Dianxing Xu, *University of Missouri - Kansas City, USA*
Han Xu, *Huawei Company, China*
Hongji Yang, *University of Leicester, UK*

James Yang, *Western Michigan University, USA*
Tao Zhang, *Macau University of Science and Technology, China*
Zhiyi Zhang, *Nanjing University of Aeronautics and Astronautics, China*
Mohammad Zulkernine*, Queen's University, Canada*

Keynote Speech 1

Faults, Failures, and Vulnerabilities: What are the Trends and How Do We Make Progress?

Dr. D. Richard Kuhn, ***IEEE Fellow, Associate Editor of IEEE Transactions on Reliability, Computer Security Division, National Institute of Standards and Technology, Gaithersburg, Maryland, USA***

Abstract

Studying trends in system vulnerabilities shows some degree of progress, but it is slow. At the same time, we see bugs ranging from user annoyances to major security flaws in much consumer software. How are ordinary bugs and security vulnerabilities related Vulnerability studies show that about two thirds of vulnerabilities are the result of ordinary coding errors, not security-specific flaws. But other studies have found little or no correlation between bugs and later vulnerabilities. How can both lines of research be correct? If most vulnerabilities are caused by regular coding errors, why isn't there a stronger correlation between the two? In this talk, we'll show that there really is no conflict, and the explanation points to ways of reducing ordinary errors as well as security flaws.

About the Speaker

Rick Kuhn is a computer scientist in the Computer Security Division of the National Institute of Standards and Technology and is a Fellow of the Institute of Electrical and Electronics Engineers (IEEE). He has authored three books and more than 150 papers on information security, empirical studies of software failure, and combinatorial methods in software testing, and co-developed the role based access control model (RBAC) used worldwide. Before joining NIST, he worked as a software developer with NCR Corporation and the Johns Hopkins University Applied Physics Laboratory. He received an MS in computer science from the University of Maryland College Park.

Keynote Speech 2

Human-Machine Pair Programming: An Intelligent and Automated Approach for Software Productivity and Reliability

Professor Shaoying Liu, ***IEEE Fellow, BCS Fellow, Associate Editor of IEEE Transactions on Reliability, School of Informatics and Data Science, Hiroshima University, Japan***

Abstract

Pair programming is one of the promising techniques advocated in agile development paradigm, but it tends to be more costly than one person-based programming and to lack a rigorous principle for governing the cooperation of the two programmers. In our recent work on Agile Formal Engineering Methods, we put forward a novel technique called Software Construction Monitoring and Predicting (SCMP) to study an intelligent and automatic approach to human-machine pair programming (HMPP). Its aim is to automatically, dynamically monitor the process of software construction for fault detection and to predict the possible future contents of the software towards its error- free completion based on existing programming experiences and knowledge. This research field is still in its beginning and there are many challenging issues to be addressed.

In this talk, I will first discuss the theoretical foundation and frameworks for Software Construction Monitoring and Predicting (SCMP) for HMPP. I will then discuss with examples how it can be applied to support specification construction and program construction, respectively. Finally, I will talk about some challenging issues to be addressed in the future.

About the Speaker

Shaoying Liu is a Professor of Software Engineering at Hiroshima University, Japan, IEEE Fellow, and BCS Fellow. He received the Ph.D in Computer Science from the University of Manchester, U.K in 1992. His research interests include Formal Engineering Methods for Software Development, Specification Verification and Validation, Specification-based Program Inspection, Automatic Specification-based Testing, Testing-Based Formal Verification, and Intelligent Software Engineering Environments. He has published a book entitled "Formal Engineering for Industrial Software Development" with Springer-Verlag, 12 edited conference proceedings, and over 200 academic papers in refereed journals and international conferences. He proposed to use the terminology of "Formal Engineering Methods" in 1997, and has established Formal Engineering Methods as a research area based on his extensive research on the SOFL (Structured Object-Oriented Formal Language) method since 1989, and the development of ICFEM conference series since 1997. In recent years, he has served as the General Chair of ICFEM 2017, the Chair of ICECCS Steering Committee, and a PC member for numerous international conferences. He is an Associate Editor for IEEE Transactions on Reliability and a member of JSSST and IPSJ.

A Novel Application Approach for Anomaly Detection and Fault Determination Process based on Machine Learning

Yang Hong
College of Computer Science and Technology
University of Aeronautics and Astronautics
Nanjing 211106, China
hongyang@nuaa.edu.cn

Lisong Wang
College of Computer Science and Technology
University of Aeronautics and Astronautics
Nanjing 211106, China
wangls@nuaa.edu.cn

Jiexiang Kang
Department of Software,
China National Aeronautic Radio Electronics Research Institute
Shanghai 200233, China
kang_jiexiang@careri.com

Hui Wang
Department of Software,
China National Aeronautic Radio Electronics Research Institute
Shanghai 200233, China
wang_hui@careri.com

Zhongjie Gao
Department of Software,
China National Aeronautic Radio Electronics Research Institute
Shanghai 200233, China
gao_zhongjie@careri.com

***Abstract*—Research on the fault is a systematic project, which mainly includes fault detection, identification, and fault handling. The system fault is often manifested in the form of abnormal data (anomaly). However, the traditional fault analysis, such as fault tree analysis (FTA), cannot explore the potential system anomaly. Therefore, it is necessary to combine the FTA with anomaly detection together. Currently, machine learning has been applied to the anomaly detection field widely. But simultaneously, it is unable to determine whether the anomaly is caused by the system fault. Focusing on fault detection and identification, this paper represents a novel approach that associates the application of machine learning in anomaly detection with the FTA organically to determine the characteristics of the anomaly. The effectiveness of the approach is verified through an illustrative example. Finally, we introduce the next stage of work about fault handling.**

***Index Terms*—Safety, FTA, Anomaly detection, Machine learning**

I. Introduction

When human society enters industry 4.0 (so named after the fourth industrial revolution), the frequent occurrence of accidents seriously restrict the economic development and social stability or even threaten the survival of human beings. Therefore, safety science plays an increasingly crucial role in engineering field, which contains food safety, nuclear safety, chemical safety, aviation safety, etc. For the safety attribute, from the academic perspective, we give one precise and professional definition. Safety is the minimization of risk and epistemic uncertainty associated with unwanted outcomes that are severe enough to be seen as harmful [**?**],i.e.,the probability of the system functioning correctly without any fault through a safe procedure called safety [**?**]. In the past half-century, the loss of human beings was serious due to the neglect of safety. For example, the Fukushima Daiichi nuclear disaster initiated by the tsunami in 2011 [**?**] was one example where the experts of the nuclear facility failed to foresee the meteorological factor which would result in accidents.

A. Review of system safety evaluation methods

Reliability engineering originated from the 1920s, which included system fault breeding, propagation, repair, prediction, etc. Since the 1980s, as a great development in the research of the safety field, these research results have been widely applied to communications, aviation, transportation, etc. There are a lot of classical safety assessment methods that help experts to perform reliability and safety analysis. The Failure Modes Effects and Critically Analysis (FMECA) [**?**] is one of them which takes all possible effects combinations of single component failure mode into consideration. Besides, the fault tree analysis (FTA) [**?**] is another approach that is used to determine system dependability. In the fault tree, the logical connections among faults and their causes are illustrated graphically and inductively. It means safety experts can work backward from the top event towards the leaves of the tree to determine the root cause of the top event [**?**]. Furthermore,

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00009

some models based on dependability analysis and application of FTA in MBDA generated quickly. Hierarchically Performed Hazard Origin & Propagation Studies or HiP-HOPS [**?** **?**]is one of the more advanced compositional model-based dependability analysis techniques. It can generate fault trees, FMEA tables, and perform quantitative analysis [**?** **?**].

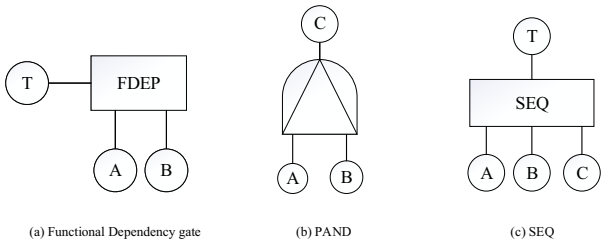
Meanwhile, countries gradually focus on the fault mode analysis to reduce the failure probability. The existing approaches of Probabilistic Safety Assessment (PSA) are classified into two parts: static probabilistic safety evaluation and dynamic probabilistic safety evaluation. The former mainly includes fault tree analysis and reliability block diagram. The dynamic safety evaluation is about the Markov transfer process (Continuous-Time Markov Chains, Semi-Markov Process, and Regenerative Process) [**?**], Numerical methods (Monte Carlo Simulations and Petri Network method) [**?**].

Based on the above analysis, we conclude that fault mode analysis such as FTA plays a vital role in fault analysis. In the engineering field, system fault is usually manifested in the form of abnormal data. However, the FTA method is unable to detect the abnormal data in the system. Meanwhile, machine learning methods which cannot analyze system fault have good effects on the anomaly detection. From this perspective, it is necessary to combine anomaly detection with fault tree analysis together. This paper presents a novel approach that associates the application of machine learning in anomaly detection with dynamic fault tree analysis organically to conduct fault analysis including determining the characteristics of anomalies and calculating the top event probability. Then we demonstrate the determining process through one illustrative example.

II. Background and related work

FTA, a top-down analysis method, was invented in 1961 in Bell Laboratories which was to help in the design of US Air Force's Minuteman missile system. After its birth, it has been applied to many fields, including the automotive, aerospace, and nuclear industries [**?** **?**]. It adopts static logic and graphical methods to express the logical relationship between the events. The fault tree defines the causes of the system failure mode or top event in terms of component failure and human errors which are represented by basic events [**?**]. Traditional static fault tree includes top event, basic evens, intermediate events, and logic gates.

As the wide application of various control and fault-tolerant techniques, many systems exhibit a variety of dynamic characteristics, such as dependency, polymorphism, and randomness. Traditional static fault tree cannot deal with the logical relationship among the components related to the dynamic characteristics, so it is impossible to analyze the system with dynamic characteristics and temporal relationship. Therefore, in 1992, Professor J.B.Dugan firstly proposed the definition of the dynamic fault tree [**?**]. The dynamic fault tree is the most prominent dynamic extension of SFTs that enable a fault tree to capture sequence-dependent dynamic behavior [**?**]. Based on the existing function of the static fault tree, new logic gates



(a) Functional Dependency gate (b) PAND (c) SEQ

Fig. 1: FDEP, PAND, SEQ

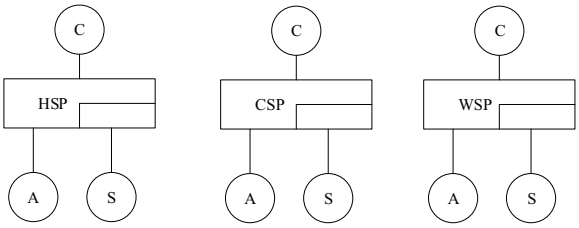


Fig. 2: HSP, CSP, WSP

including Functional Dependency gate (FDEP), Priority-AND gate (PAND), Sequence Enforcing gate (SEQ), Hot Spare gate (HSP), Cold Spare gate (CSP), and Warm Spare gate (WSP) are added. Several types of typical dynamic logic gates are illustrated in the following parts.

1) FDEP consists of a triggering event and several relevant events. In Fig.1(a), where T represents a triggering event, A and B are relevant events.
2) PAND describes the sequence of events when events must occur in a specific sequence. As shown in Fig.1(b), A, and B are input events of PAND gate, C is the output event. Event C can happen only when event A occurs before event B. Furthermore, the SEQ gate is an extension of the PAND gate. Input events must occur from left to right under one particular sequence (from A to C in Fig.1(c)).
3) Redundant backups of key performances are usually performed in the system with high-reliability requirements. Some system redundant components do not work normally only when working components fail. In that case, the backup can be divided into Hot spare, Cold spare, and Warm spare, their corresponding logic gate are respectively HSP, CSP, and WSP.

III. Contributions of our work

In the safety science domain, research work on system fault is a systematic project, which mainly divides into two parts: fault detection, identification, and fault handling. Under normal circumstances, the system fault is often manifested in the form of abnormal data. However, the FTA method cannot explore the potential system anomalies while the system is

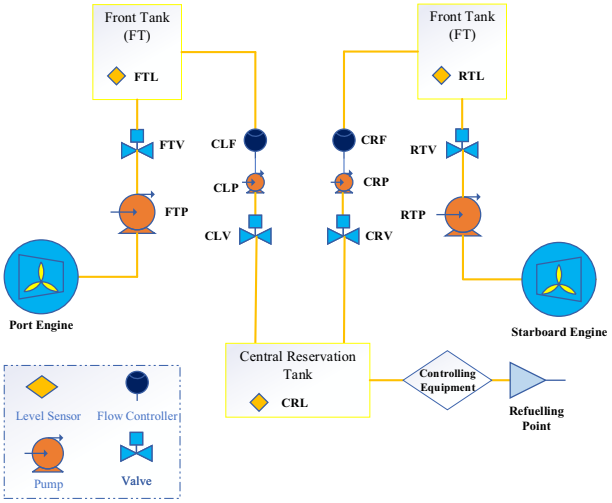


Fig. 3: Simplified structure of aviation fuel system

running. Therefore, it is reasonable to combine the FTA method with anomaly detection together. Currently, machine learning has already been applied to the anomaly detection field widely. But simultaneously, it is unable to determine whether the abnormal data is caused by the system fault.

From this perspective, for the fault detection and identification part, this paper presents a novel approach that combines the application of machine learning in anomaly detection with the fault tree analysis organically to determine the characteristics of the anomaly. Then we demonstrate the process of determining whether the anomaly is caused by system fault through an illustrative example of an Aircraft Fuel Distribution System.

Finally, if we have already determined the abnormal data is caused by the system fault, we would introduce the fault handling process for the fault handling part.

IV. The Demonstration of Anomaly Detection and Fault Determination Process

Aimed at illustrating the fault determining process intuitively, a simplified version of an Aircraft Fuel Distribution System (AFDS) in [**?**] is used. The structure of the fuel system is shown in Fig.3, which can store and distribute fuel. Obviously, the structure is symmetric, we only consider the fuel flow path of Port Engine.

We describe the fault tree of the aviation fuel system in Fig.5, E1 to E7 represent FTP.Electronic fail, FTP.Controlling valve fail, FTV Plug, Fuel pipeline Plug, FT.outlet close, FT leaked, CLP fault respectively, and I1 to I3 represent no fuel flow through FTV, no fuel flow from the front tank, the fuel level of front tank is too low respectively [**?**].

In normal condition, the Front Tank keeps providing fuel to the Port Engine, without deriving any fuel from the Central Reservation Tank. Until time interval 8, the fuel level of the Front Tank drops to 55%, where it derives fuel from the

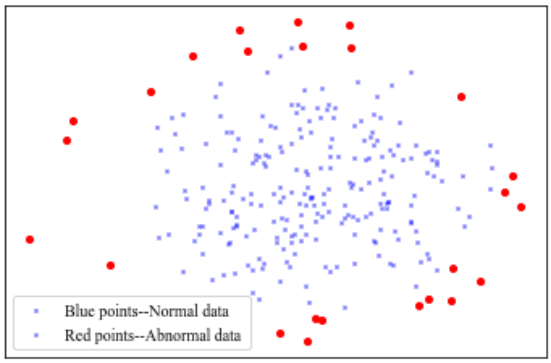


Fig. 4: Data distribution of the fuel system

Central Tank. The fuel consumption process is shown in Fig.5 (a).

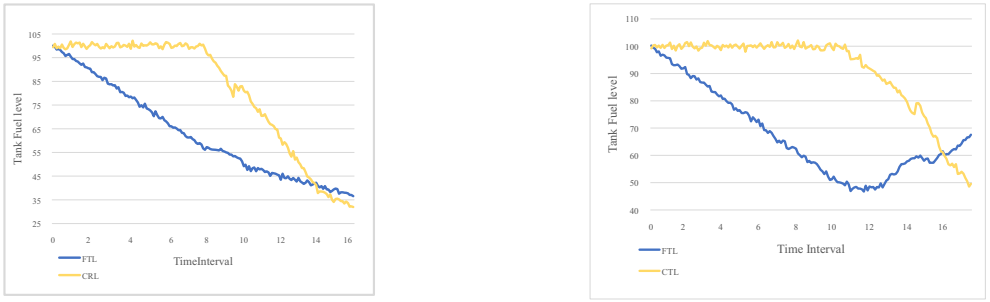
To illustrate our presented approach is feasible, we describe an abnormal condition that the fuel omission of the Port Engine exists at time interval 12. After time interval 12, the system sensors detect a cluster of abnormal data by the OC-SVM algorithm. The data distribution of the fuel system is shown in Fig.4 and the variation of fuel tank level is shown in Fig.5(b).

Then according to the fault tree created by safety experts and the calculation of component correlation parameters, we draw the conclusion that the abnormal data is caused by the omission fuel of the front tank, which is the system fault.

V. Conclusion and Future Work

In the future, our research would focus on the fault handling part. When the system fault is determined, to avoid **NP** problems, R. Gulati, J.B. Dugan [**?**] proposed a solution that we can modularize the fault tree and decompose it into independent sub-modules. In general, it is decomposed into independent dynamic subtrees and static subtrees by Depth-first Left-most(DFLM) algorithm [**?**]. Then we use the Binary Decision Diagram (BDD) method to get the minimum cut-set and calculate the probability of the top event in each static subtree. Markov transfer process can be applied to the quantitative analysis of the dynamic subtrees that represents the failure of some system components. Therefore, a fault mode can be obtained according to all transfers on the Markov transfer chain and the corresponding sequence of occurrence. We can calculate the occurrence probability of the top event in the dynamic fault tree by synthesizing the analysis results of static subtrees and dynamic subtrees. The complete analysis process is illustrated in Fig.7.

Finally, we will utilize one case study of the Integrated Modular Avionics (IMA) core processing structure to verify the effectiveness of our future work. However, one challenge of this work would be the availability of real-time operational data.



(a) Normal Condition (b) Abnormal Condition

Fig. 5: Fuel Consumption Diagram of different scenarios

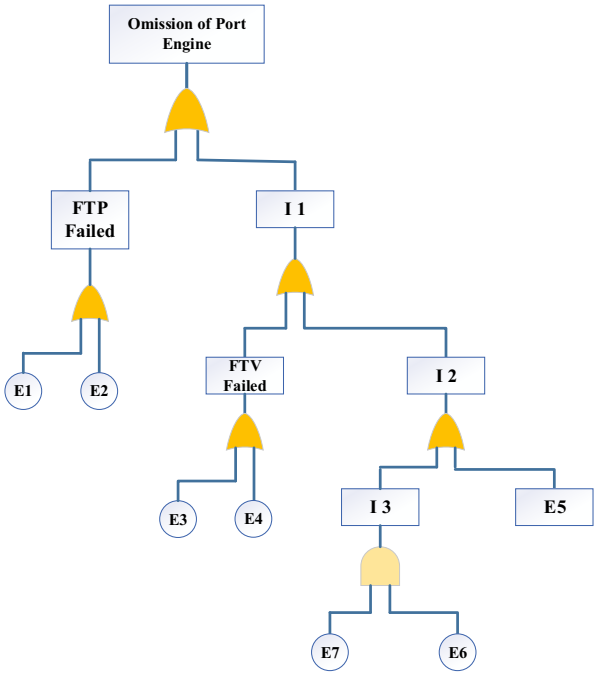


Fig. 6: The simplified fault tree of fuel system

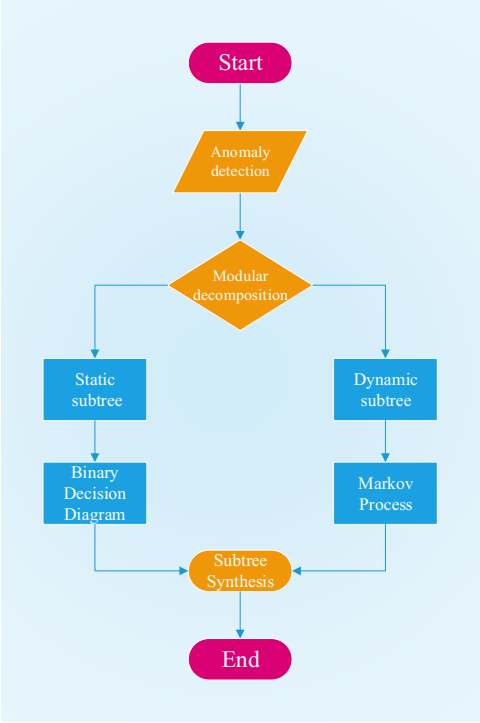


Fig. 7: The process of top event probability calculation

References

[1] N. Möller, "The concepts of risk and safety," *Handbook of risk theory: epistemology, decision theory, ethics, and social implications of risk*, vol. 1, 2012.

[2] Y. Yu and B. W. Johnson, "Safety assessment for safety-critical systems using markov chain modular approach," *International Journal of Reliability, Quality and Safety Engineering*, vol. 18, no. 02, pp. 139–157, 2011.

[3] C. Perrow, "Fukushima, risk, and probability: Expect the unexpected," *Bulletin of the atomic scientists*, vol. 1, 2011.

[4] M. Standard, "Procedures for performing a failure mode, effects and criticality analysis," *Department of Defense, Washington, DC, Standard No. MIL-STD-1629A*, 1980.

[5] W. E. Vesely, F. F. Goldberg, N. H. Roberts, and D. F. Haasl, "Fault tree handbook," Nuclear Regulatory Commission Washington DC, Tech. Rep., 1981.

[6] S. Kabir, "An overview of fault tree analysis and its application in model based dependability analysis," *Expert Systems with Applications*, vol. 77, pp. 114–135, 2017.

[7] Y. Papadopoulos and J. A. McDermid, "Hierarchically performed hazard origin and propagation studies," in

International Conference on Computer Safety, Reliability, and Security. Springer, 1999, pp. 139–152.

[8] Y. Papadopoulos and J. McDermid, "Safety-directed system monitoring using safety cases," Ph.D. dissertation, Citeseer, 2000.

[9] Y. Papadopoulos, M. Walker, D. Parker, E. Rüde, R. Hamann, A. Uhlig, U. Grätz, and R. Lien, "Engineering failure analysis and design optimisation with hip-hops," *Engineering Failure Analysis*, vol. 18, no. 2, pp. 590–608, 2011.

[10] Y. Papadopoulos, M. Walker, D. Parker, S. Sharvia, L. Bottaci, S. Kabir, L. Azevedo, and I. Sorokos, "A synthesis of logic and bio-inspired techniques in the design of dependable systems," *Annual Reviews in Control*, vol. 41, pp. 170–182, 2016.

[11] S. Distefano, F. Longo, and K. S. Trivedi, "Investigating dynamic reliability and availability through state–space models," *Computers & Mathematics with Applications*, vol. 64, no. 12, pp. 3701–3716, 2012.

[12] M. K. Molloy, "Performance analysis using stochastic petri nets," *IEEE Transactions on computers*, no. 9, pp. 913–917, 1982.

[13] S. Kabir, M. Walker, and Y. Papadopoulos, "Quantitative evaluation of pandora temporal fault trees via petri nets," *IFAC-PapersOnLine*, vol. 48, no. 21, pp. 458–463, 2015.

[14] M. Walker and Y. Papadopoulos, "Qualitative temporal analysis: Towards a full implementation of the fault tree handbook," *Control Engineering Practice*, vol. 17, no. 10, pp. 1115–1125, 2009.

[15] R. M. Sinnamon and J. Andrews, "Improved accuracy in quantitative fault tree analysis," *Quality and reliability engineering international*, vol. 13, no. 5, pp. 285–292, 1997.

[16] J. Ni, W. Tang, and Y. Xing, "A simple algebra for fault tree analysis of static and dynamic systems," *IEEE Transactions on Reliability*, vol. 62, no. 4, pp. 846–861, 2013.

[17] J. B. Dugan, S. J. Bavuso, and M. A. Boyd, "Dynamic fault-tree models for fault-tolerant computer systems," *IEEE Transactions on reliability*, vol. 41, no. 3, pp. 363–377, 1992.

[18] S. Kabir, M. Walker, and Y. Papadopoulos, "Dynamic system safety analysis in hip-hops with petri nets and bayesian networks," *Safety science*, vol. 105, pp. 55–70, 2018.

[19] Y. Gheraibia, S. Kabir, K. Aslansefat, I. Sorokos, and Y. Papadopoulos, "Safety+ ai: A novel approach to update safety models using artificial intelligence," *IEEE Access*, vol. 7, pp. 135 855–135 869, 2019.

[20] R. Gulati and J. B. Dugan, "A modular approach for analyzing static and dynamic fault trees," in *Annual reliability and maintainability symposium*. IEEE, 1997, pp. 57–63.

[21] A. Rauzy, "Some disturbing facts about depth-first left-most variable ordering heuristics for binary decision diagrams," *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, vol. 222, no. 4, pp. 573–582, 2008.

Convolutional Neural Network Algorithm based on Improved Support Vector Machine

Zhang Suzhi
School of Software
Zhengzhou University of Light Industry
Zhengzhou , China
zhsuzhi@zzuli.edu.cn

Wu Yuhong
School of Computer and Communication Engineering
Zhengzhou University of Light Industry
Zhengzhou , China
1227886042@qq.com

Abstract—**This electronic Aiming at the problem that the traditional Convolutional Neural Network algorithm uses the Softmax function to classify tires with poor accuracy, a convolutional neural network image classification and recognition algorithm based on improved support vector machine is proposed.This algorithm uses the support vector machine instead of the Softmax function to complete the image classification problem in the convolutional neural network algorithm. At the same time, a relaxation variable and a penalty factor are introduced into the traditional support vector machine, thereby changing the support vector machine to be used for classifier for multiple classification problems. This improved algorithm is applied to tire damage image recognition. Through a series of comparison experiments between the improved algorithm and the traditional convolutional neural network algorithm, the classification and recognition effects of the improved algorithm on the classification and recognition of damaged tire images are analyzed. The effectiveness of the proposed improved algorithm is proved.**

Keywords-support vector machine; image classification; convolutional neural network; relaxation variable; penalty factor

I. INTRODUCTION

In recent years, the rapid development of my country's economy has driven the rapid development of the automobile industry, causing the sales of automobiles to rise sharply. The use and compensation of automobile tires have gradually become the focus of modern society. The application of deep learning [1] in the classification and recognition of tire damaged images, the use of learning models to train a large number of tire damaged images, learning useful features, and the use of computers to automatically classify and recognize images are undoubtedly to the tire industry. New opportunities and challenges have come.

Due to the powerful learning ability of the algorithm itself [2] , convolutional neural networks are widely used in many areas of image recognition [3-5]. However, due to the large differences in image features in different fields, the training efficiency of convolutional neural networks and the accuracy of image classification and recognition are quite different. In order to solve this problem, many researchers have proposed different improvement methods. Liu Liang improved the design of the Softmax layer, added a regular term to the loss cost function of Softmax, and introduced the weight attenuation coefficient to make the test set recognition rate of the convolutional neural network in the face recognition field improve to a certain extent [6]. Wang Yuhao et al. combined the core principal component analysis method and the Softmax classification function to propose a new fault diagnosis classification model. The superiority of the model was also verified by classifying the high-voltage circuit breaker mechanical failure [7]. Zhou Fei et al. improved the classification ability of the convolutional network model by increasing the dimension of the training sample label and increasing the minimum Hamming distance of different sample categories, and combining the Sigmoid activation function with the cross-entropy loss function [8]. Aiming at the problem of low training efficiency of the Softmax layer in large amounts of data, Yang Hebiao proposed a new dynamic sequence Softmax algorithm. The node replacement method was used to dynamically construct the coding tree. The first-order moment estimation and the second-order moment estimation methods were used to dynamically adjust and update. Direction and learning rate improve the accuracy of Softmax classification in training classification for massive data [9].

In this paper, an improved convolutional neural network image classification and recognition algorithm is proposed for the problem of low accuracy of classification and recognition in the application of convolutional neural network algorithm in tire damage image recognition. By replacing the Softmax layer of the traditional algorithm with the improved support vector machine to complete the image classification, the traditional algorithm and the improved algorithm were applied to the classification and recognition of the broken tire image, and the experimental results also verified that the improved algorithm proposed in this paper improved the tire accuracy of classification and recognition of damaged images.

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00010

II. RELATED WORK

A. Convolutional Neural Network

The concept of convolutional netural networks (CNN) was proposed by LeCun in 1998, and it was successfully applied to digit recognition of handwritten fonts [10]. A general convolutional neural network consists of an input layer, a convolutional layer, a pooling layer, a fully connected layer, and an output layer. Among them, the connection of the convolutional layer and the pooling layer constitutes the image feature extraction module of the algorithm, and then the extracted feature information is transmitted to the fully connected layer. The fully connected layer integrates local feature information with classification performance and forms a new feature image [11], The output value of the last fully connected layer is passed to the output layer, which is the Softmax classification layer. This layer is equivalent to a classifier, which sets the Softmax logistic regression function to complete the image classification. The structure diagram of the convolutional neural network algorithm model is shown in Figure 1:

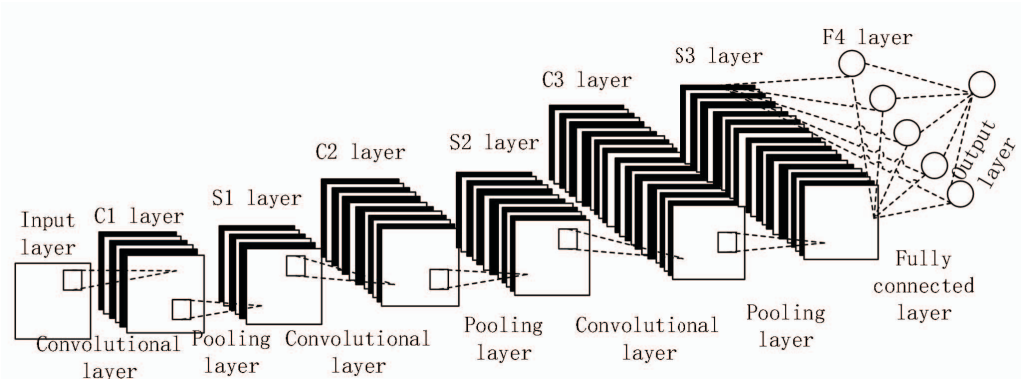


Figure 1. Convolutional neural network model structure diagram

B. Convolutional layer

The calculation expression of the convolution layer is:

$$x_j^{(l)} = f\left[\sum_{i\in M_j} x_i^{(l-1)} \otimes k_{ij}^{(l)} + b_j^{(l)}\right] \quad (1)$$

Among them, M_j represents the combination of all the input feature maps of the previous layer; $x_i^{(l-1)}$ is the i th feature map of the input tire damage image data of the $l-1$ layer; $k_{ij}^{(l)}$ is the convolution kernel, which represents the l th layer of the l layer The feature map is convolved with the i th feature map of the $l-1$ th layer; $b_j^{(l)}$ is the j th feature map of the l th layer.

C. Pooling layer

The main purpose of the pooling layer is to reduce the feature dimension of the image while keeping the number of feature maps constant, so that the calculation amount of the algorithm is reduced. Commonly used methods are average pooling, maximum pooling and random pooling. The calculation formula is:

$$x_j^{(l)} = f\left[\beta_j^{(l)} down\left(x_j^{(l-1)}\right) + b_j^{(l)}\right] \quad (2)$$

Among them, $down(\bullet)$ represents the downsampling function; $\beta_j^{(l)}$ represents the multiplicative offset after the convolution of layer $l-1$.

D. Fully connected layer

The fully-connected layer is to integrate the previous useful confidence. Through the nonlinear mapping of the activation function, the high-dimensional features are converted into low-dimensional features and output to the Softmax layer. The calculation formula can be expressed as:

$$\begin{cases} Q_i = \sum_{j=1}^{N}\sum_{k=1}^{M} x_k^j \omega_{ijk} + b_i \\ y_i = f\left(Q_i\right) \\ Q = \left[Q_1, \cdots, Q_i, \cdots, Q_m\right] \\ y = \left[y_1, \cdots, y_i, \cdots, y_m\right] \end{cases} \quad (3)$$

Among them, ω represents the weight variable; b_i represents the offset value of the

i th neuron; y_i represents the output value of the i th neuron; N represents the number of feature maps of all samples; M represents the number of entropy neurons of the feature map of sample data; m is the number of classification categories; Q represents the value of the activation function of the output layer.

E. Softmax regression classification

The classifier of the network is generally selected from three types, including logistic regression (LR) classifier, Softmax regression classifier and linear classifier. The traditional logistic regression classifier is mainly used to solve the problem of binary classification. Compared with it, Softmax regression classifier is often used to solve the problem of multi-classification. The Softmax classifier mainly uses the regression function to calculate and estimate the probability that each input data feature belongs to each image category, and maps all the classification results into a probability domain. The ultimate goal is to predict the probability distribution of the category to which the predicted data belongs and the amount of loss is minimized [12]. Assuming that the training data set is $\left\{\left(x^{(1)}, y^{(1)}\right), \cdots, \left(x^{(m)}, y^{(m)}\right)\right\}$, $x^{(i)} \in R^{n+1}$ represents the characteristics of the input data, The dimension of x is represented by $n+1$, the output data is represented by $y^{(i)} \in \{1,2,3,\cdots,k\}$, and $p = \left(y = j \middle| x\right)$ is the value of the estimated probability for each classification category j, then the loss function $h_\theta(x)$ can be expressed as:

$$h_\theta(x) = \begin{bmatrix} p\left(y^{(i)} = 1 \middle| x^{(i)}, \theta\right) \\ p\left(y^{(i)} = 2 \middle| x^{(i)}, \theta\right) \\ \vdots \\ p\left(y^{(i)} = k \middle| x^{(i)}, \theta\right) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \tag{4}$$

Among them, $\theta_1, \theta_2, \cdots, \theta_k \in R^{n+1}$ is used to represent the parameters of the model, and the calculation formula of the cost function $J(\theta)$ of the Softmax regression model can be obtained from the above formula:

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} \sigma\left\{y^{(i)} = j\right\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}\right] \tag{5}$$

Where $\sigma\{\bullet\}$ is a two-finger function, the value rule is that the value is true and the value is 1, when the value is false, the value is 0. It can be seen that the estimated probability value $p\left(y^{(i)} = j \middle| x^{(i)}; \theta\right)$ can be expressed as:

$$p\left(y^{(i)} = j \middle| x^{(i)}; \ \theta\right) = \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}} \tag{6}$$

In practical applications, it is expected that the error of the output result of the Softmax classifier is minimized, that is, the above cost function is minimized. Commonly used methods include gradient descent and L-BFGS iteration to find its optimal solution. The gradient formula can be expressed as:

$$\nabla_{\theta_j} J(\theta) = -\frac{1}{m}\left[x^{(i)}\left(\sigma\left\{y^{(i)} = j\right\} - p\left(y^{(i)} = j \middle| x^{(i)}; \theta\right)\right)\right] \tag{7}$$

III. BASED ON IMPROVED CONVOLUTIONAL NEURAL NETWORK ALGORITHM

A. SVM support vector machine

The SVM classifier is widely used in various classification predictions [13] and pattern recognition due to its global optimal, concise and flexible characteristics, and has achieved good results [14].Suppose that the data set of the tire damage image is $S = \left\{\left(x_i, y_i\right) \middle| i = 1,2,\cdots,n\right\}$, where $x_i \in R^d$ and $y_i = \{+1,-1\}$ correspond to the category labels of x_i response. Let $g(x) = \omega \bullet x + b$ denote the linear discriminant function in the d -dimensional space feature, the corresponding classification surface equation can be expressed as $\omega \bullet x + b = 0$, The discriminant function $g(x)$ is standardized, so that the image data samples of different categories all satisfy $|g(x)| \geq 1$. Under this condition, the classification interval of the samples is denoted by $2/\|\omega\|$. Now we need to do the maximum processing of the classification interval of the sample, that is, calculate the minimum value of $\|\omega\|$. At this time, the classification hyperplane is required to correctly divide all the sample data. At this time, it meets:

$$y_i\left[(\omega \bullet x) + b\right] - 1 \geq 0 \tag{8}$$

The classification hyperplane that satisfies the condition of equation (8) above is the optimal classification surface to be found. However, the problem of finding the optimal classification surface can be transformed into the problem of finding the minimum value of the objective function $\phi(\omega)$ through the calculation under the constraint of formula (8).Then the objective function $\phi(\omega)$ at this time can be expressed as:

$$\phi(\omega)=\frac{1}{2}\|\omega\|^2=\frac{1}{2}(\omega\bullet\omega) \tag{9}$$

The biggest difference between the support vector machine as the classifier and the Softmax regression classifier is that the loss functions used by the two classifiers are different. The loss function L_i of the support vector machine as the classifier can be expressed by the following formula (10):

$$L_{\mathrm{i}}=\sum_{j\neq y_i}\max\left(0,s_j-s_{y_i}+\Delta\right) \tag{10}$$

Among them, s_{y_i} represents the possibility size of the real category y_i ; s_j is the possibility size of other categories; Δ is the size of the boundary value. It can be derived from equation (10) that if the difference between correct classification and incorrect classification is greater than the value of boundary Δ, the loss function is 0, and the correction of the classification result can be stopped at this time.

B. Improve SVM support vector machine

The traditional support vector machine itself is a two-class classifier, but there are more than two types of tire damage images studied in this article, which is actually a multi-classification problem. Therefore, this paper introduces a relaxation variable ξ_i and a penalty factor C into the objective function $\phi(\omega)$ of the traditional support vector machine, then the current objective function can be expressed by the following formula (11):

$$\phi(\omega,\xi_i)=\frac{1}{2}\|\omega\|^2+C\left(\sum_{i=1}^{N}\xi_i\right)=\frac{1}{2}(\omega\bullet\omega)+C \tag{11}$$

Introduce a Lagrangian multiplier $(\alpha_1,\alpha_2,\cdots,\alpha_N)$ in the above formula (11), convert the above formula into the problem of finding the optimal classification surface under a condition with a constraint, the final value is represented by $\omega=\sum_i\alpha_i y_i x_i$, then the tire damage image The best classification function can be expressed as:

$$f(x)=\mathrm{sgn}\{(\omega\bullet x)+b\}=\mathrm{sgn}\left\{\sum_{i=1}^{N}\alpha_i y_i(x_i\bullet x)+b\right\} \tag{12}$$

C. Improved convolutional neural network algorithm

This paper studies the classification of tire damage images. The purpose is to use the computer to estimate the posterior probability of each sample x corresponding to the i category. First, calculate the class probability estimate of the tire damage image:

$$r_{ij}=p\left(y=i\,|\,y=i\text{或}j,x\right) \tag{13}$$

In the above formula, ξ_i can be obtained by solving the optimization problem given by the following formula (14):

$$\min_p\frac{1}{2}\sum_{i=1}^{k}\sum_{j\neq i}\left(r_{ij}p_i-r_{ij}p_j\right)^2 \tag{14}$$

$$s.t.\sum_{i=1}^{M}p_i=1,p_i\geq 0 \tag{15}$$

Record the largest i of p_i as $i_{\max}$. From the perspective of its posterior probability analysis, the confidence Con of the tire image sample can be defined as:

$$Con=p_{i_{\max}}-\max_{1\leq i\leq k,i\neq i_{\max}}p_i \tag{16}$$

The improved algorithm proposed in this paper replaces Softmax regression with an improved support vector machine as a classifier, classifies through the support vector machine classifier and predicts its posterior probability.

IV. EXPERIMENT AND ANALYSIS

A. Experimental environment

In order to prove the feasibility of the improved support vector machine based convolutional neural network image classification and recognition algorithm proposed in this paper, this experiment is under Windows10, 64-bit operating system, through the deep learning based on Python language to build TensorFlow platform to verify .

B. Experimental data

The experimental data set in this paper is composed of the broken tire images captured by mobile phones and crawled online, including a total of 2158 pictures. In order to ensure that the number of data samples is sufficient, the tire damage image is pre-processed before the model training, that is, the number of data samples is expanded by image pre-processing techniques

such as horizontal flip, vertical flip, random rotation or scaling technology. After pretreatment, a total of 3475 samples were finally obtained. The experiments in this paper randomly select 70% of the tire damage image samples as the data of the training set, and the remaining 30% constitute the data of the test set. The size of all tire damage images in the data set is 64*64 pixels.

C. Experimental results and analysis

1) Comparison experiment of training model

Firstly, Softmax regression and improved support vector machine are used to train and classify the broken tire image of the classification layer of the convolutional neural network image classification and recognition algorithm, and the performance of the algorithm is compared by training accuracy and loss curve. The experimental results of the algorithm training part are shown in Figure 2 below:

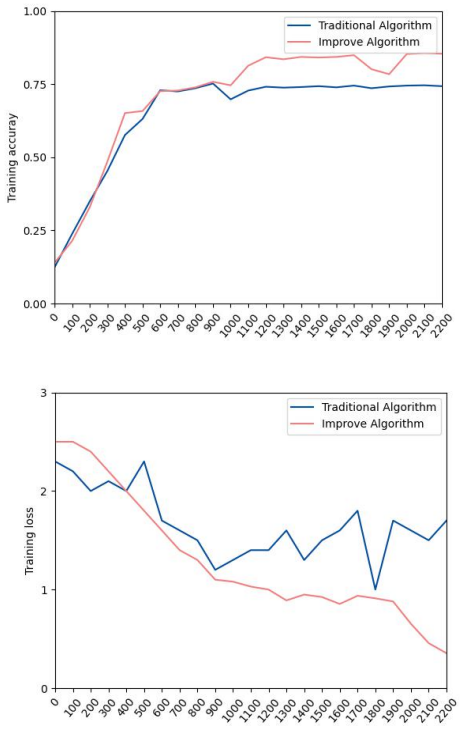


Figure 2. Comparison of training experiment results before and after improvement

In the above figure, the red curve represents the improved support vector machine convolutional neural network algorithm proposed in this paper, and the blue curve represents the Convolutional neural network algorithm using Softmax regression classification. The left picture is the comparison of training accuracy before and after algorithm improvement. As can be seen from the figure, with the continuous increase of training times, the training accuracy of both algorithms eventually tends to a relatively stable state, but the improved algorithm is accurate the rate is significantly higher than traditional algorithms. The figure on the right is the corresponding loss function curve of the algorithm. From the curve in the figure, it can also be seen that the loss of the improved algorithm is lower than the traditional algorithm, and as the number of training increases, the traditional algorithm's loss function value does not fluctuate around 1.5, but The value of the loss function of the improved algorithm gradually decreases, which also shows that the performance of the improved algorithm is better than the traditional algorithm.

2) Comparison experiment of test model

According to the tire damage image classification model obtained from the above training part experiment, Softmax regression and improved support vector machine were used to test the test set and compare the performance of the two classification algorithms. The experimental results of the test part are shown in Figure 3:

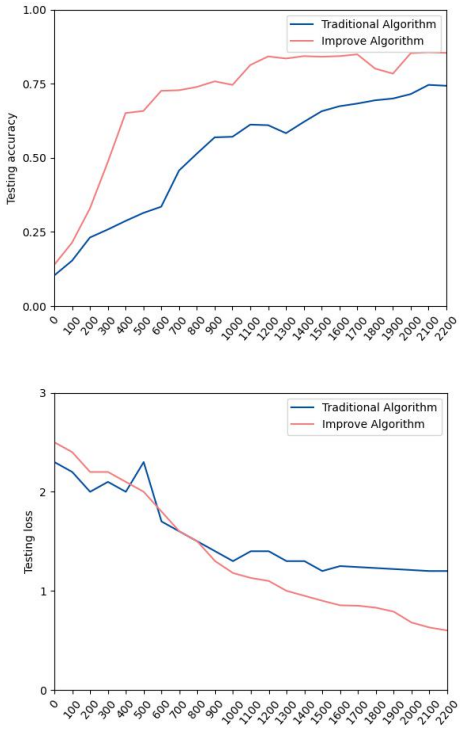


Figure 3. Comparison of test results before and after improvement

In the above experimental result graph, the left graph represents the comparison curve of the test accuracy before and after the improvement of the algorithm. It can be seen from the figure that the convolutional neural network algorithm of Softmax regression classification is used to classify and identify the tire damage image. The accuracy rate is around 70%. With the improved algorithm proposed in this paper, the accuracy of

image classification and recognition is finally close to 80%, and the accuracy of the improved algorithm has been greatly improved. The picture on the right is the corresponding loss function curve of the traditional algorithm and the improved algorithm. From the curve in the figure, it can also be seen that the overall loss function value of the improved algorithm is lower than that of the traditional algorithm. That is, in the application of tire damage image classification and recognition, it is more appropriate to use the improved support vector machine as the classifier of the convolutional neural network algorithm, and its classification performance at this time is better than the Softmax regression classifier.

V. CONCLUSION

In this paper, when the convolutional neural network algorithm is applied to the classification and recognition of tire damage images, the classification accuracy and recognition accuracy are not high enough, and a convolutional neural network image classification and recognition algorithm based on improved support vector machine is proposed. The improved algorithm firstly introduces a relaxation variable and a penalty factor based on the traditional support vector machine, and uses the Lagrange multiplier to transform the optimal classification surface, and converts the traditional support vector machine from two classification problems to Multi-classification problem. The second is to replace the Softmax regression model in the traditional convolutional neural network algorithm with an improved support vector machine to solve the image classification problem. In this paper, the performance of the classification and recognition of the improved algorithm is tested through experiments. From the experimental results, the improved convolutional neural network algorithm has improved the training accuracy and test accuracy by about 10% compared with the traditional algorithm. Therefore, in the application of tire damage image recognition, the convolutional neural network algorithm based on improved support vector machine is more suitable, the classification ability is better, and the recognition accuracy is also higher. However, the recognition accuracy of the improved algorithm is about 80%. Compared with the application of convolutional neural networks in other fields, the recognition accuracy is not too low. The follow-up work will focus on how to further improve the network model, so as to achieve the purpose of higher recognition accuracy of the tire damage image.

REFERENCES

[1] Le Cun Y.,Bengio Y.,Hinton G.(2015)Deep learning.Nature 521:436,444.

[2] Ren S , He K , Girshick R , et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6):1137-1149.

[3] Zhao X . Research on Face Recognition Based on Deep Learning[C]// Sixth International Conference on Digital Information. 0.

[4] Xin-Han L , Kan Q , Yu-Fei W , et al. Research on Character Recognition Algorithm Based on Connected Domain Detection in Natural Scene[J]. Computer Technology and Development, 2015.

[5] Nguyen T N , Le T L , Vu H , et al. A Combination of Deep Learning and Hand-Designed Feature for Plant Identification Based on Leaf and Flower Images[M]// Advanced Topics in Intelligent Information and Database Systems. Springer International Publishing, 2017.

[6] Hui L , Yu-Jie S . Research on face recognition algorithm based on improved convolution neural network[C]// IEEE Conference on Industrial Electronics & Applications. IEEE, 2018.

[7] Wang Yuhao, Wu Jianwen, Ma Suliang, et al. Mechanical Fault Diagnosis Research of High Voltage Circuit Breaker Based on Kernel Principal Component Analysis and SoftMax[J]. Transactions of China Electrotechnical Society, 2020,35(S1):267-276.

[8] Fei Z , Yang L I , Xin-Yue F . Improved Loss Calculation Algorithm for Convolutional Neural Networks in Image Classification Application[J]. Journal of Chinese Computer Systems, 2019.

[9] YANG Hebiao, HU Jingtao, LIU Fang. Training algorithm of dynamic hierarchical Softmax based on neural network language model[J]. Journal of Jingsu University(Natural Science Edition), 2020,41(01):67-72+80.

[10] Lecun Y , Bottou L . Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):P.2278-2324.

[11] Sainath T N, Mohamed A, Kingsbury B, et al. Deep convolutional neural networks for LVCSR//Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing. Vancouver, Canada, 2013: 8614-8618.

[12] Peng R , Ling W , Xin L I , et al. Improved softmax classifier for deep convolution neural networks and its application in face recognition[J]. Journal of Shanghai University(Natural Science Edition), 2018.

[13] Peng R , Ling W , Xin L I , et al. Improved softmax classifier for deep convolution neural networks and its application in face recognition[J]. Journal of Shanghai University(Natural Science Edition), 2018.

[14] Aleš Leonardis, Sanja Fidler. Learning Hierarchical Representations of Object Categories for Robot Vision[M]// Robotics Research. Springer Berlin Heidelberg, 2010.

Automatic Test Case Generation from Formal Requirement Model for Avionics Software

WenXuan Wang
College of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics
Nanjing 211106, China
wangwengx.1997@163.com

Jun Hu
College of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics
Nanjing 211106, China
hujun@nuaa.edu.cn

JianChen Hu
College of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics
Nanjing 211106, China

JieXiang Kang
Department of Software
China National Aeronautic Radio Electronics Research Institute
Shanghai 200233, China

Hui Wang
Department of Software
China National Aeronautic Radio Electronics Research Institute
Shanghai 200233, China

ZhongJie Gao
Department of Software
China National Aeronautic Radio Electronics Research Institute
Shanghai 200233, China

***Abstract*—The scale and complexity of software in modern avionics systems are growing fast. It becomes a challenge to develop such kind of software systems satisfying both high safety and reliability requirements. This paper presents a framework of automatic test case generation for avionics software form formal requirement models. We introduced a specific formal model of VRM (Variable Relation Model), which is used for modeling the requirements of avionics software, from which control tree structures are established. Different model test coverage criteria are defined according to the DO-178C standard to simplified the control tree structure, and test path constraint selection methods are proposed based on those criteria. Through performing domain error test case selection, test cases are generated for each path constraint selected, which makes up a test case set for the requirement. At last, to demonstrate how we generate test cases form a requirement model, a case study is given.**

***Index Terms*—Test case generation, Requirements-based testing, Formal requirements model, Avionics**

I. Introduction

The avionics system is a typical software-based safety-critical system. The development of modern avionics software is a time-consuming and costly process. In the development of such software projects, software verification and software validation phase (V&V) consume about 50% -70% of the software development resources [1]. V&V technology includes various types of software analysis, review and testing technologies. In comparison, software testing technology is still a widely used low-cost software verification technology in the field of avionics. With the rapid growth of the scale and complexity of avionics software, and the gradual promotion and application of model-based systems/software engineering [2], traditional avionics software testing technology is facing more and more challenges, such as: how to generate an effective set of test cases based on a requirement model automatically, and how to integrate formal methods in test technology, etc., which can not only reduce the cost of software development but also improve the quality of software testing, improve software safety and reliability.

From the perspective of system engineering and software engineering, a complete, consistent and well-organized avionics software requirements product is the core of improving avionics software quality and reducing development costs. In the DO178-C "Software Considerations in Certification of Airborne Systems and Equipment" issued by RTCA [3], it emphasizes the development of various development activities with multi-level requirements as the core and defines test standards for different safety-critical software. And in the latest DO-333 annex [6], formal methods [7] are introduced to supplement and replace the traditional avionics software analysis and verification technology. The formal models built in the requirements stage can provide accurate requirements semantic information for subsequent system analysis and effective generation of test cases.

The work of this paper proposes an automated generation method of software test case technology based on the formal requirements model in the field of avionics. Among them, the formal requirements model is the avionics Requirements model (VRM: Variable Relations Model) based on the formal table model established in other projects related to the work of this article. Based on the requirements semantic model accurately defined by VRM, we analyzed the path structure in the VRM requirements model, get the control tree structures of the requirement model, and defined model test coverage criteria according to the test standards for different safety-critical software specified in the DO-178C. For each path constraint, we propose an error based test point select method to found out the test points in these paths that were most likely

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00011

to find typical errors of the software.

The rest of the paper is organized as follows: Section II outlines the relevant background knowledge of this work, including DO-178C avionics software standard and model-based system/software engineering(MBSE: Model-Based System/Software Engineering). Section III describes the overall framework of the project description and automatic test case generation method of this work. Then in section IV, the semantic mapping relationship between the model elements of the formal requirements model VRM and the control tree structures is given, and to reduce the number of test cases three model test converge criteria are defined. Section V introduces the method of the domain error based test case selection strategy. Then in section VI, the approach is illustrated by an example.

II. BACKGROUND

A. Avionics Software and DO-178 (B/C) Standard

Because the Avionics system is a typical safety-critical system, compared to errors introduced during the design or implementation phase, errors in avionics software requirements are more likely to have an important impact on the safety of these critical systems. Therefore, in the DO-178B "Software Considerations in Certification of Airborne Systems and Equipment" issued by RTCA in 1992, the focus of avionics software development was given with processes and objectives as the core concepts. Among them, system requirements, High-level software requirements and low-level software requirements are top priorities in avionics software development. In 2012, DO178B was upgraded to DO-178C, adding support for high-level modeling (DO-331 standard), object-oriented (DO-332 standard), and formal methods (DO-333 standard), and further introduced the two-way traceability mechanism with requirements as the core. Formal methods are a type of mathematical techniques used to develop computer hardware and software systems. They are usually divided into three major types of methods: model checking, proof of the theorem, and abstract interpretation. The rigor of mathematical methods can support model analysis at different levels involved in the life cycle of computer system R&D projects. Especially in the stage of requirements and architecture design, formal methods can be used to accurately acquire, explain and describe requirements and their corresponding constraints. The Requirements model VRM used in the work of this paper is based on the core semantics of the four-variable model [8] and is a formal model after domain-oriented tailoring for avionics software. Its detailed description will be given in section IV.

B. Model-based system/software engineering (MBSE)

Model-based system/software engineering (MBSE) is a type of methodology for the design and implementation of complex systems and software that has been applied and promoted in the industry in the past decade. The core idea of MBSE is to establish system/software models at different development stages (such as requirements, design, architecture, implementation, etc.), which can effectively carry out model-based analysis, conversion and verification at different levels, thereby reducing development errors, reduce costs and improve development efficiency. The application models at different levels are many and many types, the most common is the system modeling language SysML [9]. At present, in the field of avionics systems at home and abroad, MBSE methods and tools have been widely used at the system architecture design level, such as SCADE [10], Rhapsody [11], etc. Model-based software development standards (DO331) were also introduced in DO178C. The work of this paper belongs to the category of automatic test case generation based on the Requirements model under the MBSE overall framework.

C. Related work

Blackburn and Busser [4] developed the T-VEC system, which uses theorem-proving methods to generate test cases. Specify valid system properties as logical formulas. In the process of proving the formula, the tool generates a test vector with specific values for input and output. Then, an input sequence that leads the system to the state of interest can be obtained by explicitly constructing a formula that constrains all states in the sequence. Simon Burton of York University [5] also discussed the use of theorem proving methods in generating test cases.

Ammann, Black and Majurski proposed a novel mutation-based method in which SMV was used to generate test offspring sequences. By applying mutations to both specifications and attributes, they can obtain a large number of test sequences. The correct software implementation should pass the description of the correctly executed test and the incorrectly executed failure test. Callahan uses a process that represents the specification to check the trace generated by the process of the simulation program. In this way, they can detect and analyze differences between software implementations and specifications (a set of attributes). Engels et al. also described the method of generating test sequences using Spin.

III. TEST CASE GENERATION METHOD FRAMEWORK FOR AVIONICS SOFTWARE REQUIREMENTS MODEL

In this section, first of all, the project background of the work content of this article and the relationship between the work of this article and other project work are given. Then give the overall process description of the test case automatic generation technology based on the VRM formal requirements designed in this paper.

The content of this work is part of the research content of a complete project ART (Avionics Requirement Tools)(see Fig. 1). The goal of the entire project is to design and implement a software tool platform (ART) for requirements analysis and verification in the DO-178C standard for the need of the DO-178C standard. In the stage of Requirements modeling, considering the domain characteristics in the requirements of modern civil aircraft avionics software, with the formal method of "four-variable theoretical model" as the core, a practical engineering project containing elements such as events, conditions, patterns, and environmental interactions

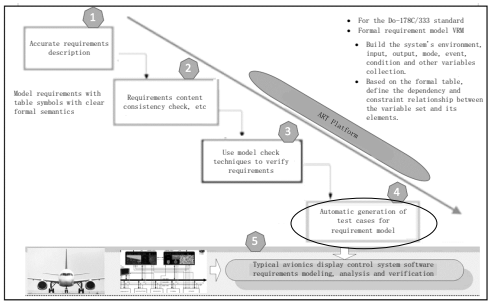


Fig. 1. Avionics Requirements Modeling, Analysis ,and Verification Tool Platform (ART).

was constructed. Formal Variable Relationship Model (VRM). Then, based on the formal semantics of VRM, the analysis and verification of consistency and completeness are carried out. Finally, the formalized VRM requirements model can be used to automatically generate test case sets.

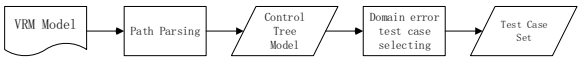


Fig. 2. Overview of Test Case Generation Methods Based on the VRM Requirements Model.

Fig. 2 further reveals the basic framework of the work in this paper, that is, the automatic generation of test cases based on the VRM Requirements model. It is divided into several major steps. First, before starting the test case generation of the requirements model, it must be preprocessed, which is the path division and the control tree structure generation. Then, the appropriate branch is selected in the control tree species path, for each path, perform error sensitivity analysis, and domain error test case selection. Finally, integrate all branch tests according to the analysis results form a collection of test cases based on the requirements model.

IV. Requirements model preprocessing

In this section, the definition and examples of VRM are given first, and then the relationship between the VRM requirement model and the requirement implementation code structure is described. Based on this, the VRM model control tree structure and test coverage criteria are proposed.

A. VRM:A Formal requirements model for avionics

Similar to the data storage and processing methods in relational databases, the intuitive form of the VRM model is to use a two-dimensional table to build a requirement model. The two-dimensional relational data in the relational database is actually a mathematical model strictly defined by the relational calculus logic system (ie: formal model). The VRM is a type of four-variable model that tailored according to the characteristics of the avionics software domain. Its formal definition is as follows:

A VRM requirement specification is defined by a six-tuple:$\{SV, C, E, F, TS, VR\}$,Where SV is the set of all state variables, it is a four-tuple, defined as:$SV = \{MV, CV, M, IV\}$,Including supervision variable MV, controlled variable CV, mode class M, intermediate variable IV. The function of each data of the six-tuple is specifically described below.

- MV: non-empty set of disjoint monitor variables, $MV = \{mv_1, mv_2, \ldots, mv_i\}$, Where $mv_1, mv_2, \ldots, mv_i$ is called the monitor variable.
- CV: Non-empty disjoint set of controlled variables, $CV = \{cv_1, cv_2, \ldots, cv_j\}$,Where $cv_1, cv_2, \ldots, cv_j$ is called the controlled variable.
- M: Non-empty disjoint collection of model classes, $M = \{mc_1, mc_2, \ldots, mc_m\}$,Where $mc_1, mc_2, \ldots, mc_m$ is called the model. mc_k is a pattern class, which contains all the models under the model class, $mc_k = \{mc_{k_1}, mc_{k_2}, \ldots, mc_{k_n}\}$.
- IV: Non-empty disjoint set of intermediate variables, $IV = \{iv_1, iv_2, \ldots, iv_k\}$, Where $iv_1, iv_2, \ldots, iv_k$ is called the intermediate variable.
- TS: Union of types, where all types are non-empty sets of values.
- VR: A special function used to map the name of a state variable to a specific value, indicating all value ranges of the state variable. For all $r \in VR(r)$ in VR, r is a variable in SV, and TS is the range type of r. $VR(r)$ is the set of possible values of r.
- C: Condition, indicating a predicate on a single state variable, such as $Altitude > 500$ means the current height is greater than 500. The condition is a logical expression, with multiple expressions, which can be a boolean variable true, false, or a boolean expression $c_i \bigodot c_j$, etc. $\bigodot \in \{AND, OR, NOT\}$ represents logical operator; $C = r \circ v$. Where $\circ \in \{=, <, >, \neq, \geq, \leq\}$ represents the relational operator.
- E: Event, representing the predicate on two state variables, the general expression of the event is

$$EVENT(S) \textbf{ GUARD } D.$$

  $EVENT \in \{@T, @F, @C\}$ represents the event operator; $GUARD \in \{WHEN, WHERE, WHILE\}$ represents the guard operator.
- F: table function, all tables are a mathematical function, all can be expressed by F_i.

There are three categories of table functions in the VRM model: condition table, event table, and mode conversion table. All three types of tables have corresponding formal semantic definitions. Due to space limitations, the following only gives a brief description with examples in Table I, which is an example of a condition table. The semantics are: Based on the state dependency $D_n = \{Pressure, Overriden\}$, the value of the controlled variable $SafetyInjection$ is defined

(that is, a functional requirement F_6); the corresponding two-dimensional table is intuitive Models representing models and corresponding mathematical logic formulas.

TABLE I
EXAMPLE OF CONDITION TABLE

Mode Pressure	Condition	
High,Permitted	True	False
TooLow	Overridden	Not Overridden
SafetyInjection	Off	On

the corresponding mathematical logic formulas as follows:

$$SafeInjection = F_6(Pressure, Overridden) =$$
$$\begin{cases} Off, & if\ Pressure = High \vee Pressure = Permitted \\ & \vee (Pressure = TooLow \wedge Overridden = true). \\ On, & if\ Pressure = TooLow \wedge Overridden = true. \end{cases}$$

In the avionics system, the system requirements must meet the two properties of completeness and atomicity. The former requires that the requirements contain all the information needed to implement the system, which means that we can learn All possible behaviors of the system. The latter requires requirements to be independent. Based on this, our testing of requirements-based software systems can be broken down into tests for each requirement, and each entry in the table of the VRM model is a clearly defined requirement.

B. The control tree structure of a VRM model

Hoare [12] uses preconditions/postconditions to represent the requirement model, as shown in Figure 5. A precondition is an input space vector. Similarly, a postcondition is an output space vector. Association predicates group all the prerequisites (input space vectors) associated with each functional relationship. Each association predicate describes the data and timing constraints on the input spatial object. Functional relationships describe objects in the output space as functions of the input.

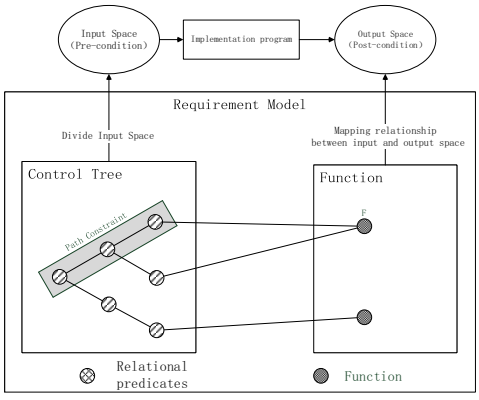


Fig. 3. Pre/post-condition representation of the requirement model.

There may be many forms of requirements implementation, but all programs that implement correct implementations should have corresponding control tree structures that are equivalent to statement implementation and requirement models. To better describe the control tree results of the requirement model, we make the following divisions of the elements in the VRM requirement model.

- **Model Decision**: The condition for each variable in the output variable table function of the VRM requirement model is called a decision.
- **Model condition**: a single logical predicate in each decision.
- **Model statement**: each assignment of output variables, intermediate variables, and model table functions.

As in the table function F_6 of the output variable $SafetyInjection$ in Table. I, The model elements are divided as shown in Table. III.

TABLE II
MODEL ELEMENT DIVISION

Type		
Model Statement	**Model Decision**	**Model Condition**
$S = On$	$A\&B$	$A : Press = TooLow$
		$B : Overwrite = false$
$S = Off$	$C \mid D \mid (A\&\neg B)$	$C : Press = High$
		$A : Press = TooLow$
		$\neg B$

[a]To reduce the layout, S is used here to represent $SaftyInjection$

In the process of avionics software development, for the correct requirement realization program, a model decision corresponds to a series of decision statements in the program, model conditions correspond to the conditional expressions in these decision statements, and model statements correspond to a series of calculation or assignment statement, which execution result should be equivalent to a model statement.

Here, we propose the definition of the control tree structure of the requirement model:

- **Basic block**: Corresponds to a model statement.
- **Control tree**: It can be represented by a triple, that is $T = (N, E, n_0)$, where N represents all the nodes sets in the control tree, the node set contains only two parts, one part is the basic block, The other part is the model decision. If the model decision contains multiple model conditions, split the multiple conditions into a single condition, and use "6.A" to represent the condition A in the table function F_6, and E represents all the directed in the control tree. Edge set, n_0 represents the entry node of the model.
- Path constraint: It is a set of inequalities or equations from the entrance node to a leaf node in the control tree, which can be represented by a tuple $P_i = (p_i, f_i)$, where $p_i = \{C_{i1}, C_{i2}, \ldots, C_{ij}\}$, $f_i = \{f_{i1}, f_{i2}, \ldots, f_{ik}\}$. C_{ij} represents the model condition j on the path p_i, and the conditions are all single conditions; The set of all variables that satisfy the constraints of P is called the

domain $D(P)$ of the path constraint P, the domain D(p) is bounded by the domain boundary represented by the relation predicate C of path p. $C(P)$ represents a model statement on the path p, p corresponds to some control flow statements in the program, that is, an execution path, And f corresponds to a series of calculation assignment statements in the control flow in the program.

The control tree structure for table function F_6 in Table I is as shown in figure 4.

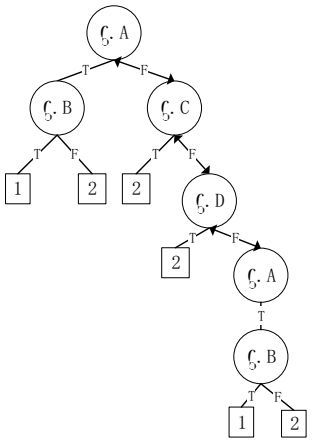


Fig. 4. The control tree structure for table function F_6

The control tree structure contains six path constraints. The two-way arrow connecting line indicates that the two conditions are mutually exclusive.

The test based on the requirement model is based on this control tree structure. We will test each path constraint in the control tree, select the test points that satisfy the path constraints, and solve the expected output of each test point.

C. Definition of requirement model test coverage criteria

For complex requirement models, the number of branches of the control tree is very large, resulting in many path constraints, especially in the case of many model conditions, it will grow exponentially. And the repeated sub-tree structure will appear in the control tree, as shown in Figure. 4, the repeated sub-tree structure appears. If all the path constraints in the control tree are tested, it will result in too many test path constraints, the test case set is too large, and there are many redundant tests.

To reduce the test scale and redundant testing, and improve test efficiency, we propose a coverage criterion [13]–[16] based on the VRM requirement model, as follows:

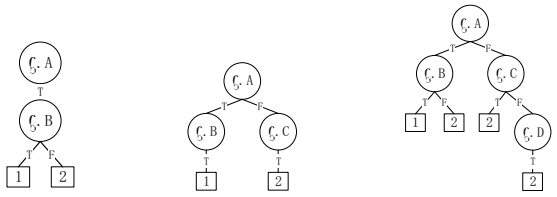
- **Statement coverage for the VRM model**: Statement coverage for the VRM model: The control tree path for generating test cases covers all model statements.
- **Path coverage for VRM model**: The control tree path for generating test cases covers all model decisions.
- **MC/DC coverage of the VRM model**: The control tree path for generating test cases satisfy the statement and path coverage for VRM model, and:
  - All possible results of any model decision appear at least once.
  - All possible results of each model condition determined by any model appear at least once.
  - Each condition can independently affect the result of the corresponding decision, that is, there are two paths, where the model decision result corresponding to the condition is opposite, the value of the condition is opposite, and the other conditions in the model decision where the condition is located have the same value.

Tests that satisfy the model statement coverage can find that the implementation of the output function mapping is incorrect in the implementation of the requirements, such as overflow, underflow, or incorrect implementation. The test that meets the model decision coverage can find the coverage of the division of various values of the requirement table function in the requirement realization. The test covering the MC/DC coverage of the VRM model can effectively reduce the number of test cases generated by the full path constraint coverage, and can effectively check the logical operator errors in the model decision, or incorrectly program the variable or expression to their negative form.

D. Test path constraint selection based on test coverage criteria of the VRM model

in this summary, we give a method to simplify the control tree structure based on the test coverage criteria of three requirement models. And the simplified control tree structure that the table function F_6 obtained by this method meets the coverage criterion is given.

a) Model statement coverage based test path selection: For the statement coverage, traverse the requirement model control tree and find a path containing the statement for each statement in the model. As shown in Figure 3, the control tree is traversed sequentially to obtain two path constraints: $(6.A)\rightarrow(6.B)\rightarrow(1)$ and$(6.A)\rightarrow(6.B)\rightarrow(2)$. The simplified control tree is shown in figure 5(a).



(a) (b) (c)

Fig. 5. The simplified control tree structure for F_6

b) Model decision coverage based test path selection: For each model decision, search for the condition node in the control tree that involves the decision, select the left subtree to find a path, and then find a path in the right subtree. For each model decision to perform the same operation. Finally, delete the duplicate path. As shown in Figure. 4, the test path constraints that satisfy the decision coverage are: $(6.A) \rightarrow (6.B) \rightarrow (1)$ and $(6.A) \rightarrow (6.C) \rightarrow (2)$. The simplified control tree to be tested is as shown in figure 5(b).

c) Model MC/DC coverage based test path selection: for each decision, list the decision value and condition value table first. For table function F_6 the model decision and condition is shown in table IV.

TABLE III
MODEL DECISIONS FOR TABLE FUNCTION F_6

No	A	B	$A\&B$
1	T	T	T
2	F	NG	F
3	NG	F	F

No	A	C	D	B	$C \mid D \mid (A\&\neg B)$
4	NG	T	NG	NG	T
5	NG	NG	T	NG	T
6	T	NG	NG	F	T
7	NG	F	F	T	F

NG indicates that the decision result has nothing to do with this condition, so whether the value of this variable is true or false will not affect the true and false decision. After obtaining this table, traverse each condition, looking for a pair of path constraints where the value of the condition is different and the other conditions are the same, and the decision result is opposite, where NG can be either T or F. Merge all paths, remove duplicate paths, The simplified path table is shown in table V, and the simplified control tree is shown in figure 5(c).

TABLE IV
SIMPLIFIED PATH TABLE BASED ON MODEL MC/DC

No	**A**	**C**	**D**	**B**
4	F	T	F	NG
5	F	F	T	NG
6	T	F	F	F
7	T	F	F	T

V. TEST CASE SELECTION STRATEGY

In this section, the classification and definition of code implementation errors concerned in the field of avionics are first given, including two major categories, namely: path errors and calculation errors. Then, the definition of the error sensitivity of the test case set is given, and finally, the test case selection strategies [17]–[19] based on path error and calculation error are given respectively.

A. Two types of code implementation errors

In Section IV, we proposed the control tree structure of the requirement model and apply the model coverage criterion to obtain the set of path constraints to be tested. For each path constraint P, we need to analyze its model conditions and model sentences to select appropriate test cases to test it.

For each path-constrained test case selection, here are two extreme solutions:

- Randomly select a test point in the domain D of P to obtain a test case set with only one test case.
- Select all test points in the domain D of P to generate a set of test cases containing all values in D.

The first solution has little effectiveness, with a low possibility of finding errors, but high feasibility because of the low cost of generating test case sets. The latter is contrary to it. As long as the program makes an error on this path constraint, it will be tested. However, in most instances, the feasibility is very low, for the cost of generating and using test case sets may be extremely high. It may be impossible for complex path constraints.

The principle of selecting test points for a certain path constraint is to use as few test points as possible while ensuring test quality. Therefore, we should select the test points that are most sensitive to errors in the implementation code for testing.

To find the test points that are most sensitive to errors, we classify and define the errors that may occur in the implementation of requirements. The path constraint is represented by the (P, F) binary, and there may be two types of errors in the implementation code. P errors are model condition implementation errors, and F errors are model statement implementation errors.

a) Path Error: Suppose P is the wrong requirement realization used to calculate the function f, and P^* is the correct requirement realization. Assume that there is an isomorphism between the path P^i of P and the path P^* of P^*, so that for all path pairs (P^i, P^*), $C(P^i) = C(P^*)$, But for some pairs (P_k, P_k^*), $D(P_k) \neq D(P_k^*)$, at this time, the requirement realization P contains a path error.

The path error is caused by the shift of the domain boundary. There are three cases of the domain boundary shifts are shown in figure 6.

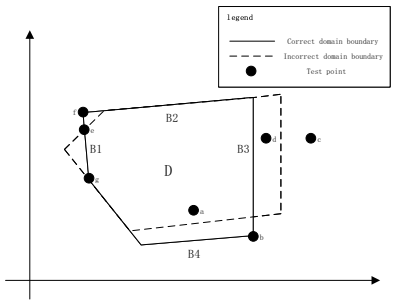


Fig. 6. Three kinds of domain errors

In figure 6, the solid line represents the domain scope of a certain path constraint P in a VRM model, and the dotted

line represents the domain scope corresponding to the path constraint in the incorrect implementation code. The domain boundary B_4 offset reduces the domain $D(P)$. Test point a can still produce the correct output because although the boundary B_4 has changed, it is still in the correct domain $D(P)$. However, the boundary B_4 has moved past the test point b, causing it not to be in $D(P)$. Since the program will follow the wrong execution path when executing input b, incorrect results will be produced. As boundary B_3 moves, domain $D(P)$ expands. Since the test point c here is still outside the domain $D(P)$, it will be processed correctly, but the test point d can detect the shift of the domain boundary, which should be outside the domain $D(P)$. Finally, among the test points e, f, and g, only test point f will be incorrect since the border shift causes it to be out of $D(P)$.

b) Calculation error: Suppose P is the wrong program for calculating function F, and P^* is the correct program. Suppose there is an isomorphism between the path P^i of P and the path P^* of P^*, so that for all path pairs (P^i, P^*), $D(P_k) = D(P_k^*)$, but For some pairs (P_k, P_k^*), $C(P^i) \neq C(P^*)$, at this time, the program P contains calculation errors.

B. Test point selection strategy based on error analysis

When a domain error occurs in a program, it appears as a domain boundary shift, so we use the domain boundary shift to describe the severity of the domain error. The greater the domain boundary shift, the more serious the domain error. The slighter domain error means that most test points cannot detect the error. Therefore, we propose the concept of error sensitivity to describe the ability of test points to detect minor domain errors. The stronger the ability to detect domain errors, the more likely it is to detect minor domain errors [20]–[22]. Specific definitions are given below.

Domain error sensitivity $S(p)$: When the test point p can be detected, the lower the minimum shift d of the domain boundary, the higher the domain error sensitivity of the test point. That is, $S(p) = 1/d$, the higher the sensitivity, the greater the probability that the test point detects a domain error.

Therefore, it is clear that points closer to the domain boundary are more sensitive to domain errors. Point b located on the boundary of the domain has the highest sensitivity to domain errors on the boundary. It should be noted that in Figure 6, the intersection b of the two domain boundaries is highly sensitive to the offset of the two domain boundaries B_4 and B_3, while the point e located on the domain boundary is more sensitive to domain errors. Point a in the domain, so the final domain error sensitivity ranking is $S(b) > S(e) > S(a)$.

we can draw the following conclusions: in the domain, the sensitivity of the test points to domain errors from high to low: domain vertices, points on the domain boundary except for the domain vertices, test points near the domain boundary, and test points inside the domain [23].

a) Path error test case selection strategy: The value of the test points on the domain vertices is the highest. For a given domain expression $D = \{L_1, L_2, \ldots, L_n\}$, even if it is the simplest n-dimensional linear domain space, the number of domain vertices is 2^n, so it is almost impossible to get all the domain vertices, so here we use a heuristic algorithm to get some domain vertices.

1) First, we divide the logical expressions in the domain expression into three categories:
   - **Basic items:** the logical relationship between an input variable and a constant, such as $x < 0, y > 9$, etc.
   - **Basic clause:** the logical relationship between a set of variables or functions and constants, such as $x + y > 10, cos(x) > 0.5$, etc.
   - **Complex expressions:** the relationship between variables, the left and right of the relationship symbol are not constants, such as $x + y > z$, etc.
2) Use the basic items, basic clauses, and complex expression initialization fields in sequence to use the value range of the variables.
3) Select the upper bound of one of the variables, and replace the variable in all non-basic term logical expressions with the upper bound of this value. This step is similar to the operation of the lower bound value.
4) Re-constrain other variables and repeat operation 2-3.
5) When the values of all variables are determined, a domain vertex test point is obtained.

VI. Case study

We demonstrate our method through a simple example, in which we first formalize the given requirements while using the programming language to implement the requirements, then use the method to generate test case set T, finally, we modify the programming implementation, deliberately inject the error set E before use the test case set T to test the program that has injected the error $e \in E$, and analyze the test results.

A. A VRM requirement model

Table VI gives a simple text form requirement in the field of avionics, where the inputs are the values of A and B sensors and the output is C.

TABLE V
A TEXT FORM REQUIREMENT

Requirement	Remarks
There are two sensors A and B in the system. If the value of sensor A or B is greater than or equal to 0, The value of port c is set to the largest value of A and B, otherwise it is 0	The type of A, B, C is Short integer

The VRM model includes two monitoring variables $MV = \{A, B\}$ a controlled variable $CV = \{C\}$, the mode and intermediate variables are empty, and the condition is $C = \{c_1, c_2, c_3\}$,where $c_1 = A > B, c_2 = B > 0, c_3 = A > 0$,, The table function F_1 is shown in table VII.

TABLE VI
TABLE FUNCTION

Mode	Condition		
	$A \geq B\& A \geq 0$	$Not(A \geq B)\& (B \geq 0)$	$Not(A \geq 0 \mid B \geq 0)$
C	A	B	0

B. Control tree generation and test path extraction

We apply the method in Section IV to generate a control tree structure for this table function F_1. The control tree structure of F_1 is shown in figure 7.

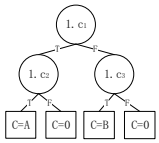


Fig. 7. The control tree structure of F_1

.

Using model coverage criteria to simplify this control tree structure, the results are shown in figure 8.

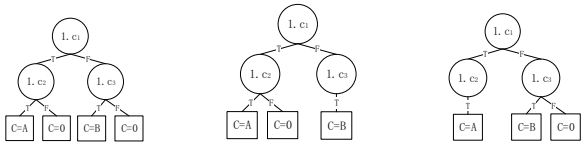


Fig. 8. Simplified control tree structure for F_1

C. Test case selection

We choose the simplified control tree shown in figure 8(a) for test case extraction, and its path constraints are shown in Table VIII.

TABLE VII
PATH CONSTRAINT TABLE

No	Path	Calculate	Domain
1	$A \geq B\& A \geq 0$	$C = A$	$A \in [0, 32767]$, $A \geq B$, $B \in [-32768, 32767]$
2	$B > A\& B \geq 0$	$C = B$	$B \in [0, 32767]$, $A + 1 \leq B$, $A \in [-32768, 32767]$
3	$A \geq B\& A < 0$	$C = 0$	$A \in [0, 32767]$, $A \geq B$, $B \in [-32768, -1]$
4	$B > A\& B < 0$	$C = 0$	$B \in [-32768, -1]$, $A + 1 \leq B$, $A \in [-32768, -2]$

For each path constraint, we extract the domain error test case at its vertices. As shown in figure 9, the vertices for each path constraint are as follows.

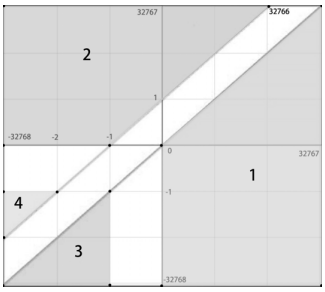


Fig. 9. Domain range of each path constraint

- For Path Constraint 1: $(0, 0)$,$(32767, 32767)$, $(32767, -32768)$,$(0, -32768)$.
- For Path Constraint 2: $(-1, 0)$,$(32766, 32767)$,$(-32768, 0)$,$(-32768, 32767)$
- For Path Constraint 3: $(-1, -1)$,$(-32768, -32768)$,$(-1, -32768)$.
- For Path Constraint 4: $(-2, -1)$,$(-32767, -32766)$,$(-32767, -1)$.

So we get a set of test cases, as shown in Table IX.

TABLE VIII
TEST CASES FOR TABLE FUNCTION F_1

No	Path	Output	Input	
		C	A	B
1	1	0	0	0
2	1	32767	32767	32767
3	1	32767	32767	-32768
4	1	0	0	-32768
5	2	0	-1	0
6	2	32767	32766	32767
7	2	0	-32768	0
8	2	32767	-32768	32767
9	3	0	-1	-1
10	3	0	-32768	-32768
11	3	0	–1	-32768
12	4	0	-2	-1
13	4	0	-32767	-32766
14	4	0	-32767	-1

D. Test results

We have written a correct implementation program for this demand, and based on this program, modify its code to get some incorrect program. The correct and modified codes are as follows:

The correct implementation:

```
short PostiveMax(short a,short b){
        short max = a>b?a:b;
        return max>=0?max:0;
}
```

TABLE IX
MODIFIED PROGRAM

No	Modified line	Modified code
1	2	**short** max = a<b?a:b;
2	2,3	**short** max = a<b?a:b; **return** max<=0?max:0;
3	2,3	**short** max = a<b?a:b; **return** max>=90?max:0;
4	2,3	**short** max = a<b?a:b; **return** max>=-100?max:0;
5	2,3	**short** max = a<b?a-1:b+1; **return** max>=0?max:90;
6	2,3	**short** max = a<b?a:b; **return** max>=0?max:0;

For these implementation codes, we use test them with test cases in table IX. The results are show in figure XI, $\surd$ means test passed, $\times$ means test failed.

TABLE X
TEST RESULT

Code	0	1	2	3	4	5	6
Error Type	null	D	D	D&C	D&C	D&C	C
Case	Test Result						
1	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\times$
2	$\surd$	$\surd$	$\times$	$\surd$	$\times$	$\times$	$\surd$
3	$\surd$	$\times$	$\times$	$\times$	$\times$	$\times$	$\surd$
4	$\surd$	$\surd$	$\times$	$\surd$	$\times$	$\surd$	$\times$
5	$\surd$	$\surd$	$\times$	$\surd$	$\times$	$\times$	$\times$
6	$\surd$	$\times$	$\times$	$\times$	$\times$	$\times$	$\surd$
7	$\surd$	$\surd$	$\times$	$\surd$	$\times$	$\times$	$\times$
8	$\surd$	$\times$	$\times$	$\times$	$\times$	$\times$	$\surd$
9	$\surd$	$\surd$	$\times$	$\surd$	$\times$	$\surd$	$\times$
10	$\surd$	$\surd$	$\times$	$\surd$	$\surd$	$\surd$	$\times$
11	$\surd$	$\surd$	$\times$	$\surd$	$\surd$	$\surd$	$\times$
12	$\surd$	$\surd$	$\times$	$\surd$	$\times$	$\surd$	$\times$
13	$\surd$	$\surd$	$\times$	$\surd$	$\surd$	$\surd$	$\times$
14	$\surd$	$\surd$	$\times$	$\surd$	$\surd$	$\surd$	$\times$

[a] For the error type, null means no error, D means domain error, and C means calculation error.

According to the test results in table XI, the test case set generated by the method we proposed can effectively detect the errors that may occur when the requirements are realized.

VII. SUMMARY

This paper describes a method for generating test cases by analyzing the formal requirements model during the development of avionics software. The structure tree analysis method of demand model and three requirement model test coverage criterions are proposed. We introduced a method of selecting test points based on implementing error sensitivity analysis. Finally, we demonstrated how to use this method to generate test cases for a formal requirements model. The final test results reveal that the set of test cases generated by this method can effectively find various possible errors in the realization of requirements.

REFERENCES

[1] Rayadurgam S , Heimdahl M P E . Test-Sequence Generation from Formal Requirement Models[C]// IEEE International Symposium on High Assurance Systems Engineering. IEEE, 2001.

[2] Franz T , Lüdtke, D, Maibaum O , et al. Model-based software engineering for an optical navigation system for spacecraft[J]. 2018.

[3] Brosgol Benjamin. Do-178c[J]. ACM SIGAda Ada Letters, 2011.

[4] M. R. Blackburn, R. D. Busser, and J. S. Fontaine. Automatic generation of test vectors for SCR-style specifications. In Proceedings of the 12th Annual Conference on Computer Assurance, COMPASS'97, June 1997.

[5] S. Burton, J. Clark, and J. McDermid. Testing, proof and automation. An integrated approach. In Proceedings of First International Workshop on Automated Program Analysis, Testing and Verification, June 2000.

[6] Eisemann U , Allen J L . New Requirement-Definition and Verification Techniques According to DO-178C, DO-331, and DO-333[C]// AIAA Infotech @ Aerospace. 2016.

[7] Ingo Brückner, Heike Wehrheim. Slicing an Integrated Formal Method for Verification[M]// Formal Methods and Software Engineering. Springer Berlin Heidelberg, 2005.

[8] Hu A J , Vardi M Y . [Lecture Notes in Computer Science] Computer Aided Verification Volume 1427, SCR: A toolset for specifying and analyzing software requirements[J]. 1998, 10.1007/BFb0028725(Chapter 50):526-531.

[9] Friedenthal, Sanford. A practical guide to SysML: the systems modeling language / 2nd ed[M]. Morgan Kaufmann, 2015.

[10] François Xavier Dormoy. SCADE 6 A Model Based Solution For Safety Critical Software Development[J]. Proceedings of European Congress on Embedded Real Time Software, 2008.

[11] Chao T, Junwei F, Ling X, et al. Application of MBSE Method During Landing Gear System Design for Civil Aircraft[J]. Civil Aircraft Design & Research, 2015.

[12] Hoare, C. A. R., An axiomatic basis for computer programming, Communications of the ACM, 12(10):576583, October 1969.

[13] Reed, Greg. Design and test converge.(Brief article)[J]. Test & Measurement World, 2006(August).

[14] Zheng Y L , Ma L H , Zhang L Y , et al. On the convergence analysis and parameter selection in particle swarm optimization[C]// International Conference on Machine Learning & Cybernetics. IEEE, 2003.

[15] Yue-Hua D , Xin F U , Xiao-Ning Z . Convergence Analysis of FEM Model Based on Semi-circular Bending Test[J]. Western China Communications ence & Technology, 2012.

[16] Jianmin Wang, Xiaohua Wang, Yunyun Ma, et al. Hierarchical Combination Design Method of Test Cases Based on Conditional Constraints[C]// IEEE International Conference on Software Quality. IEEE, 2017.

[17] Lu B , Songrong Q , Gendu A Z . AN AUTOMATIC CONFORMANCE-TEST CASE GENERATION METHOD BASED ON FINITE-STATE MACHINE[J]. JOURNAL OF COMPUTER RESEARCH AND DEVELOPMENT, 1996.

[18] Jing-Li W U , Song-Feng H , Hai-Nan D . Integration test case generating method based on UML[J]. Computer Engineering and Design, 2008.

[19] Zhao-Yun S , Jun Z . A UML 2.0 Activity Diagram Based Test Method for Web Services[J]. shandong science, 2010.

[20] Mayrhauser A V , Mraz R , Walls J , et al. Domain based testing: increasing test case reuse[C]// Computer Design: VLSI in Computers and Processors, 1994. ICCD '94. Proceedings. IEEE International Conference on. IEEE, 1994.

[21] Ravi Prakash Verma, Bal Gopal, Md. Rizwan Beg. Generation of test cases from software requirements using[J]. 2014.

[22] Andres Nötzli, Khan J , Fingerhut A , et al. p4pktgen: Automated Test Case Generation for P4 Programs[C]// the Symposium. 2018.

[23] Nikoleta A , Yannis S , Fausto B , et al. A DSM Test Case Applied on an End-to-End System, from Consumer to Energy Provider[J]. Sustainability, 2018, 10(4):935.

DADF: A Dynamic Adaptive Method for Generating Adversarial Examples

Zhiwen Jiang, Zhanqi Cui, Yiting Zheng, Jiao Deng, and Xiulei Liu
Computer School
Beijing Information Science and Technology University
Beijing, China
2017010736@mail.bistu.edu.cn, czq@bistu.edu.cn, 2017011321@mail.bistu.edu.cn,
2018010478@mail.bistu.edu.cn, liuxiulei@bistu.edu.cn

***Abstract*—Deep neural networks (DNNs) have made remarkable achievements in several areas, such as image classification. However, extensive researches show that DNNs are vulnerable to adversarial examples which are generated by adding some small perturbations on original images. DeepFool is an effective algorithm for generating adversarial examples with higher success rate and smaller perturbations. However, DeepFool computes the distance of all classification boundaries of the dataset, which causes the speed of DeepFool to be very slow. To address this problem, this paper proposes DADF, a dynamic adaptive method for generating adversarial examples. Based on tracing the information of the attacking process, DADF dynamically selects the subset of target labels. The subset of target labels is dynamically updated once successful adversarial examples are generated. Experimental results on MNIST and CIFAR10 datasets show that, comparing with DeepFool, DADF reduce the time greatly while generating 10000 adversarial examples on the MNIST and CIFAR10 dataset.**

***Index Terms*—adversarial examples, robustness, neural network**

I. Introduction

Deep neural networks (DNNs) have solved complex problems effectively and produced valuable results in people's daily life, such as image processing [1] [2], object recognition [3] [4] [5], image classification [6] [7], etc. However, DNNs are vulnerable to adversarial examples with small perturbations which cannot be observed by human. Particularly, such perturbations can fool DNNs to output error results easily with a high confidence. This issue draws attentions of researches to generate the adversarial example since it helps to identify the vulnerability of the models before deploying [8]. In addition, the adversarial examples can also be used to attack DNN models.

In 2013, Szegedy et al. [9] firstly found an intriguing phenomenon of DNNs that using a certain hardly perceptible perturbation can cause DNNs to misclassify an image and proposed the concept of adversarial examples. After that, Goodfellow et al. [10] proved the existence of adversarial examples and gave the first approach of generating adversarial examples. Recently, many researches of generating adversarial examples are based on the gradient of DNNs or the distance between adversarial images and original images. These methods can be summarized as gradient-based methods and optimization-based methods. The one-step gradient based attack methods, like FGSM [10], are traditional gradient-based methods. These methods have a very fast generation speed. But the generated examples exhibit low adversarial success rate and weak robustness. The optimization-based attack methods, like DeepFool [11], perform well in adversarial success rate and robustness. However, these methods are required to solve an optimization problem such as finding the smallest perturbation of the input image data. Particularly, in order to find the smallest perturbation, DeepFool needs to traverse all the label sets and choose the closer labels as the target to add some perturbations. DeepFool is relatively slow in the process of generating adversarial examples because the time are largely consumed for computing the distances between current label and other labels in the label sets. Adversarial training [12] [13] [14] is a very effective method to defense adversarial attacks. Adversarial training using adversarial examples as a part of training dataset to train the DNNs. In this case, we need to generate adversarial examples with high qualities in a short time.

In this paper, we propose DADF (Dynamic Adaptive DeepFool method), a dynamic adaptive method based on DeepFool framework to accelerate the generation of adversarial examples. First of all, based on the distribution of adversarial examples, DADF reduces the number of target labels to be traversed. Then, by computing the distance between the input image data and the selected subset of target labels, DADF choose the label with the shortest distance to minimize the loss function [11]. Experiments are carried out on the MNIST and CIFAR10 datasets, the results show that DADF has higher success rate and more robustness than DeepFool, but when generating the same number of adversarial examples, DADF cost less time than DeepFool.

The main contributions of this paper can be summarized as follows:

- We use DeepFool to generate adversarial examples on the test dataset and give the distribution of adversarial examples and found an interesting phenomenon —- different

This work is supported in part by the Qin Xin Talents Cultivation Program of Beijing Information Science Technology University (No.QXTCP C201906) and 2020 Promote the Interior Development of University - Student Research and Training Project of Beijing Information Science and Technology University (5102010805).

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00012

Algorithm 1 Dynamic Adaptation method

Input: a dictionary $D : \{i : [j]_1^k (j \neq i)\}_1^k$, l: number of the target labels, p: label of current input data.
Output: RL: subset of target labels.

```
RL ← {}
for each label in l do
    pl ← choose a label from D[p] randomly
    RL ← RL + pl // add label pl to RL
    D[p] ← D[p] \ pl // remove label pl from D[p]
endfor
return rl
```

subsets of adversarial examples have similar distributions for specific labels.

- We present DADF method to save the time costs of generating adversarial examples. The generation process is faster than DeepFool, because DADF dynamically choose the target labels based on the information of the attack process.
- We evaluate DADF on the MNIST and CIFAR10 datasets and the experimental results show that DADF not only achieves higher attack success rate and robustness but also saves the time greatly.

The rest of paper is organized as follows. Section 2 presents our method to generate adversarial examples. Section 3 carries out experiments and compares the performance between DADF and DeepFool. Section 4 summarizes the related work of generating adversarial examples. Section 5 concludes the paper and discusses the future work.

II. The Dynamic Adaptive DeepFool Method

In this section, we elaborate the proposed the DADF algorithm. Firstly, recording the adversarial attack information of each success adversarial examples. Secondly, selecting the target labels based on the recorded adversarial information. Finally, applying selected target labels to DeepFool framework to generate adversarial examples. Specifically, for each type of the label, selecting a fixed number of labels to be traversed for generate adversarial examples. At the same time, updating the adversarial information in order to make the next choice more accurate.

A. The Dynamic Adaptation Method

Dynamic adaptative label set generation is a sub module of DADF. The main task of this module is to find specific subset of label sets according to the attack information dictionary and current input label. The detail of the Dynamic Adaptation method (DA) is showed in Algorithm 1.

Let $D : \{i : [j]_1^k (j \neq i)\}_1^k$ be a dictionary computed using our DADF method in section 2.1. Let i be the key of dictionary D between 1 and k. The value of key i is a list with value between 1 and k but except value i. Here, k is corresponds to the k_{th} class. Keys of D is the label set of origin dataset. The value of D is a list containing the value of the label set of origin dataset. l is the length of specific subset. Label p is current image data label, and it is one of the keys in D. The D is initialized once with each value of label set. Firstly, initializing RL to an empty set. Then, DA starts l iterations, in every iteration DA uses the follow method to select a different label than before and update it to RL. $D[p]$ is the source label list, it records all the adversarial attack information in the process of generating success adversarial examples for the key p. When we selected a label pl from $D[p]$, we will remove the other value in $D[p]$ that equals to pl.

D records the adversarial attack information for all the label set in origin dataset in the process of generating success adversarial examples. For example, for MNIST dataset, all the label set is 0-9, so the value of all keys in D is 0-9. By constantly updating D in the process of generating adversarial examples, we can get the list of all successfully labels with specific key, and the count of each label can be regarded as the possibility of the attack target label. The attack information on possibilities can be used to guide the selection of target labels.

B. Integrating DA to DeepFool

In this subsection, we introduce how to integrate the DA method to DeepFool and the method of generating a batch of adversarial examples effectively using Dynamic Adaptation DeepFool (DADF) method.

DeepFool is an optimization-based method and the perturbations are smaller than FGSM method, the l^∞ robustness of DeepFool are also smaller on MNIST and CIFAR-10 datasets [11]. DeepFool is an effective attack method, even many defense methods are not performing well on DeepFool attack. DeepFool performed well in adversarial attack, but how about time efficiency of DeepFool? We measure and compare the adversarial time efficiency by how much time the attack method costs for generating adversarial examples based on fixed number input data. To find the closest target label to generate adversarial examples, DeepFool traverses all the labels of the dataset and then computing the distance of each classification boundary, choosing the closest classification boundary to add some perturbations in gradient direction. Computing the distance of each classification boundary costs a lot of time, some classification boundary even far away from current input data. DADF is proposed to reduce the number of the traversed labels by taking advantage of the adversarial attack information in the process of generating adversarial examples. In this way, DADF reduces the time for generating a large number of adversarial examples. At the same time, since the adversarial information is used, it can ensure that the generated adversarial examples are similar to those generated by the original DeepFool algorithm.

In clustering problem, input data with the same attributes can be automatically clustered into a group by trained clustering algorithm. Most of the distance measures of clustering are based on Euclidean distance. Conversely, we can assume that different input images classified by neural network into the same label should also have some relevance. Images are

transformed into vectors when it is used as the input of neural networks, so if two images are classified by neural network into the same label. The distances between the two vectors should be relatively close, and the classification boundary of the trained neural network also has high similarity. Through a large number of experiments and to the law of large numbers, we can find out the possible classification boundary between the data classified as a specific label by the neural network. In this way, we can use the closer classification boundaries to calculate the gradient when generating the adversarial examples. Therefore, those less likely to be successfully attacked labels for current input image label do not need to calculate the gradient. Only several labels which likely to be successfully attacked for current input image need to calculate the gradient. To compare with traversing all labels of the dataset, it can reduce the times of calculating gradient, and can also improve the efficiency of generating adversarial examples.

The detail of DADF method is showed in Algorithm 2. In DADF algorithm, firstly the dictionary D is initialized that keys are the label sets and the value of each key is a list containing the one label of label set expect current key. For an image data x in dataset X, using DNN models to predict the label, if the predicted label is not equal to the original label, we do not apply the image data to generate adversarial examples. Secondly, y is the label of x and l is the number of specific subset labels , in line 4 of Algorithm 2, put the params D, y, l to the DA function and the function will return a label set with length l. DA choose the label randomly. The value of D with specific key is a list contain all the labels attacked to the key, so if we randomly choose an label from the label list, the possibility is positive correlation with most likely being successfully attacked label in other angle. Line 5 and line 11 is part of DeepFool framework and this part is used to generate perturbation and add it to original image data. Finally, apply DeepFool framework with T iterations to generate adversarial examples. Especially, if the input data's label doesn't change after T iterations, we think it is attacking failed. SAE is used to record the success adversarial examples. Once DADF successfully attacks the neural network model, the success adversarial example will be added to SAE and dictionary D will be updated by appending original label to value list while the key of value list is equals to the attack success label. The selection of l is very important because it controls the traversed label number of DADF algorithm. If the value of l is too small, it cannot ensure the success rate in the confrontation process, because it is less likely to get the closest label to the current input data and it may not be able to across classification boundaries in T iterations and will lead to the failure of generating successful adversarial examples. If l is too large, it will cost much more time. l is between 1 and $||K|| - 1$, $||K||$ is the length of label set, especially, when l is equal to $||K|| - 1$, it is the same with DeepFool algorithm. T is the parameter that controls how many iterations the DADF algorithm executed and marks the current adversarial attack as a failure if it has not been successfully attacked the model. The selection of T has a great influence on the time and the accuracy of generating adversarial examples. If the value of T is too large, it needs longer time to generate adversarial examples. If the value of T is too small, it will lead to a lower attack success rate of adversarial examples.

Algorithm 2 Dynamic Adaptive DeepFool method

Input: f: a classifier, J: loss function; X: dataset, Y: the label set; T: iterations, l: number of the target labels.
Output: SAE: success adversa rial examples.

1: **Initialize** a dictionary $D : \{i : [j]_1^k (j \neq i)\}_1^k$, $ASE \leftarrow \{\}$
2: **for** (x, y) in (X, Y) and $\hat{k}(x) = y$ **do**
3: **Initialize** $x_0 \leftarrow x$
4: **for** $t = 1$ to T **do**
5: **for** k in DA(D, y, l) ll_p **do**
6: $w'_k \leftarrow \nabla f_k(x_t) - \nabla f_{\hat{k}(x_0)}(x_t)$
7: $f'_k \leftarrow f_k(x_t) - f_{\hat{k}(x_0)}(x_t)$
8: $\hat{l} \leftarrow \arg\min_{k \neq \hat{k}(x_0)} \frac{|f'_k|}{\|w'_k\|_2}$
9: **endfor**
10: $x_{t+1} \leftarrow x_t + r_t$
11: **IF** $\hat{k}(x_{t+1}) \neq \hat{k}(x_0)$ THEN
12: $ASE \leftarrow ASE + \{x_{t+1}\}$
13: $D[y] \leftarrow D[y] + \hat{k}(x_{t+1})$
14: **break**
15: **endfor**
16: **endfor**
17: **return** SAE

In DADF algorithm, we propose to select a subset of labels to reduce the number of computing distance. To record the trace of adversarial labels in the process of attacking the neural network model can improve the probability of successfully choosing the right label as the target. If the selection process is performed well, we can greatly improve the efficiency of DeepFool in generating a batch of adversarial examples.

III. Experiments and Evaluations

In this section, we carry out experiments on several DNNs to compare the effectiveness of DADF with DeepFool. In addition, we give the distribution of adversarial examples generated by DeepFool.

A. Setup

The experiments are carried out on Centos7 with 8 cores Intel Core i7 CPU without CUDA support and 16G memory. The DNN models trained by the MNIST and CIFAR10 datasets respectively. The MNIST and CIFAR10 are popular datasets which are often be used to generate adversarial examples. The DNN models are trained on the MNIST and CIFAR10 datasets based on the LeNet model. The model trained with MNIST dataset has three fully connected network and handles image data with only one channel, outputs the possibility of 10 labels. The model trained with CIFAR10 dataset has the

same architecture with LeNet model and handles image data with three channels, also output the possibility of 10 labels.

Attack Success Rate. Applying all image data to generate adversarial examples, but some adversarial examples might fail to attack. So, in our experiments, the success rate of attack is defined as:

$$ASR = \frac{X_s'}{X_a'} \tag{1}$$

$X_s^{'}$ is defined as the rate of adversarial examples which can successfully attack DNN models. Precisely, the adversarial example should satisfy $f(x_{adv}) \neq y$ while y is the label of original input image and $f(x_{adv})$ is the output label with adversarial example. x_{adv} is the adversarial example of the original input data x. $X_a^{'}$ is defined as all the adversarial examples generated in the process of the attack.

Average Robustness. We denote $RobAvg(f)$ as the average robustness of successful adversarial examples generated from a classifier f, which is computed by Eq (2).

$$\mathrm{RobAvg}(f) = \frac{1}{\|N\|} \sum_{x \in N} \frac{\|r(x)\|_2}{\|x\|} \tag{2}$$

In Eq (2), $r(x)$ is the estimated minimal perturbation value generated by the generation algorithm of adversarial examples. x is the original input data. N denotes the size of dataset we used. The test datasets of MNIST and CIFAR10 dataset are used in our experiments and $||N||$ represents the number of the test dataset.

B. Distribution of Adversarial Examples

The original label and adversarial label distribution on all test dataset are reported in Table 1 and Table 2. The success adversarial examples are generated on the model trained with the MNIST and CIFAR10 dataset. The attack method is DeepFool of iteration 3. In Table 1 and Table 2, the value of cell(i, j) shows that the number of success adversarial examples, Original Label(O) is i and Adversarial Label (A) is j.

TABLE I
THE DISTRIBUTION OF ADVERSARIAL EXAMPLES GENERATED BY DEEPFOOL ON MNIST

O A	0	1	2	3	4	5	6	7	8	9
0	0	0	12	0	2	0	54	34	4	37
1	5	0	7	1	11	5	6	13	9	3
2	37	50	0	94	39	1	6	184	33	3
3	3	9	49	**0**	0	45	3	36	51	56
4	0	54	2	0	0	1	20	42	30	110
5	10	8	0	100	1	0	81	8	68	103
6	48	6	1	0	29	47	0	0	44	2
7	4	100	53	20	8	**0**	**0**	0	2	36
8	44	35	59	86	7	38	43	18	0	58
9	3	11	1	4	113	9	0	**519**	31	0

From Table 1 and Table 2, we can observe that the distribution of attack labels has a certain rule for a specific label of image data. Some labels account for high proportions, while other labels account for low proportions. Based on this, we can assume that it has a preference for attacking to specific target labels for the sample with a fixed label. This assumption can also be easily confirmed by people's feelings. For example, people are easily to regard 9 as 1, rather than 5 or 2 on the MNIST dataset. So, by using this attack information like origin labels and adversarial labels in the process of generating adversarial examples, we can select some target labels that the current input image data has higher probabilities to successfully generate adversarial examples of these labels. According to the previous attack information matrix to choose labels rather than all labels, we can reduce the number of reversed labels.

TABLE II
THE DISTRIBUTION OF ADVERSARIAL EXAMPLES GENERATED BY DEEPFOOL ON CIFAR10

O A	0	1	2	3	4	5	6	7	8	9
0	0	84	153	42	185	14	26	95	304	83
1	131	0	37	31	35	**11**	39	77	147	484
2	115	17	0	152	288	91	153	127	32	16
3	15	22	104	**0**	190	277	173	146	20	42
4	35	13	204	116	0	62	251	271	27	11
5	19	14	99	**408**	122	0	80	216	19	14
6	25	45	135	257	339	56	0	64	19	52
7	47	30	99	134	409	**138**	**48**	0	17	70
8	386	171	45	45	70	21	17	32	0	207
9	125	**409**	**50**	48	33	**23**	38	**148**	119	0

Further analysis from Table 1 shows that for a specific label, only 4-6 labels have a large proportion of the successful generation of adversarial example labels for all the original labels, while the proportion of other categories of labels is very small. Subsequent experiments also show that the adversarial examples generated for these specific labels are more easier to attack the models.

C. Comparison on Attack Success Rates

TABLE III
THE ATTACK SUCCESS RATES (ASR) OF DEEPFOOL AND DADF ON LENET

Classifier	T	ASR			
		DeepFool		DADF	
		MNIST	CIFAR10	MNIST	CIFAR10
LeNet	3	30.09%	98.10%	98.82%	99.00%
LeNet	4	71.24%	98.29%	99.33%	99.15%
LeNet	5	93.28%	98.32%	99.37%	99.36%
LeNet	6	98.27%	98.34%	99.35%	99.38%
LeNet	7	99.39%	98.36%	99.33%	99.42%
LeNet	8	99.63%	98.36%	99.38%	99.47%
LeNet	9	99.7%	98.38%	99.43%	99.42%
LeNet	10	99.75%	98.38%	99.47%	99.46%

Table 3 shows the success rates of DeepFool and DADF on MNIST and CIFAR10 dataset. The adversarial examples are generated on MNIST dataset and CIFAR10 dataset via using DeepFool, DADF attack models with iterations between 3 and 10. These experiments use all the test dataset (10000 images) as the input data, but the input data apply to the experiment is

actually 10000 multiply by the accuracy of the neural network model. The other experiments in this paper use all these test datasets as input images.

This experimental result of Table 2 shows that DADF has higher attack success rate in every iteration, and even in less iterations case like iteration number equals 3, DADF can achieve 98.82% attack success rate on MNIST dataset. On CIFAR10 dataset, DADF also has a little more higher attack success rate than DeepFool. In general, DADF performs higher attack success rate than DeepFool.

D. Comparison on Average Robustness

TABLE IV
THE AVERAGE ROBUSTNESS OF DADF AND DEEPFOOL WITH ITERATIONS BETWEEN 3 AND 10

Classifier	T	RobAvg			
		DeepFool		DADF	
		MNIST	CIFAR10	MNIST	CIFAR10
LeNet	3	0.664	0.147	0.217	0.129
LeNet	4	0.933	0.148	0.220	0.117
LeNet	5	1.055	0.149	0.219	0.118
LeNet	6	1.085	0.149	0.218	0.118
LeNet	7	1.094	0.149	0.221	0.122
LeNet	8	1.095	0.149	0.219	0.121
LeNet	9	1.095	0.149	0.218	0.118
LeNet	10	1.094	0.149	0.220	0.121

Table 4 shows the Average Robustness (RobAvg) of the adversarial examples generated by DADF and DeepFool with iterations between 3 and 10. The smaller the value of RobAvg, the better robustness of the adversarial perturbations of the network. From Table 4, the RobAvg value of DADF is smaller than DeepFool, so DADF can generate the more robustness adversarial examples than DeepFool.

E. Comparison on Attack Efficiency

TABLE V
COSTING TIME IN THE PROCESS OF ADVERSARIAL ATTACK USING DADF AND DEEPFOOL WITH ITERATIONS BETWEEN 3 AND 10

Classifier	T	Cost Time(s)			
		DeepFool		DADF	
		MNIST	CIFAR10	MNIST	CIFAR10
LeNet	3	4227.2	1639.73	657.08	470.94
LeNet	4	5239.97	1586.13	666.03	499.83
LeNet	5	5609.08	1630.88	673.33	595.96
LeNet	6	5664.69	1526.29	683.55	567.83
LeNet	7	5705.67	1550.45	672.19	535.45
LeNet	8	5649.17	1421.16	675.38	538.15
LeNet	9	5726.59	1585.24	677.73	481.72
LeNet	10	5786.43	1510.98	681.18	620.49

Table 5 shows the time costs in the process of adversarial attack using DADF and DeepFool. When iteration in the range of 3 and 5, the cost time of DeepFool increase fast but DADF increase slow. In the case of iteration number behind 5, the cost time of DADF and DeepFool almost unchanged. In general, DADF is faster than DeepFool and DADF nearly cost 1/6 times than DeepFool on MNIST dataset.

To emphasize DADF indeed learned the distribution of DeepFool, now we show the distribution of adversarial examples generated by DADF in Table 6. Compare with table 1, for a special label, the adversarial examples generated by DADF has nearly the same count as DeepFool. So, even though we decrease the labels of target label set, DADF could able to find the better adversarial perturbations to generated adversarial examples.

The results demonstrate that DADF is faster than DeepFool, because DADF indeed learns the distributions of success adversarial examples. Besides, DADF achieves higher attack success rate and more robustness than DeepFool.

IV. RELATED WORK

In this selection, we give the background knowledge of adversarial attack. Adversarial attack can be categorized into black-box attack methods and white-box attack methods according to the environment. We introduce their basic attack method as follows.

A. Black-box Attack Methods

The black-box attack method [15] means that attacker can't access to the model settings or model parameters and therefore they are unable to perform back propagation on the network they attacked, they only know the input data and the output labels of DNN. In this case, it needs more experiences to build the attack model. The most classic proposed approaches are query-based method. Query-based approaches attack the model by query the DNN repeatedly [16] [17].

B. White-box Attack Methods

By using white-box attack method to attack DNNs, all the information in DNNs even backpropagation for gradient can be used to enable the attack. For a specific DNN model, we can attack in a special way, so it performs better than the black-box attack method in many respects including success rate. In the literature, gradient-based method and optimization-based method are most often used in the research, so we briefly summarize a few important white-box attacks below.

C. Gradient-based Methods

FGSM [10]. proposed the Fast Gradient Sign Method (FGSM) to generate adversarial examples quickly. They add perturbation that computed by gradient of the model with a loss function J using a sign function on it only once. The distortion constraint $||\delta||_\infty \leq \epsilon$ is under L_∞. Assume the original input is x_0, the real label is y and the step size in the direction of gradient is ϵ, FGSM generate a adversarial example x^* by: $x^* = x_0 + \epsilon \cdot sign(\nabla J(x_0, y))$.

I-FGSM [18] is also a gradient-based method, it is an iterative version of FGSM where it uses small step size in each iteration. In each iteration, FGSM is used as the core approach the result after clipping is the input data of next iteration.

D. Optimization-based methods

In optimization-based method, the attacker generates adversarial examples by minimize a loss function. Such method can cost a lot of time but sometimes can generate more robustness adversarial examples.

L-BFGS [9] szegedy used a box-constrained L-BFGS to minimize the l_2 norm of the perturbation subject to $f(x+\delta) = l^{'}$ ($l^{'}$ is the predict label after adversarial attack while the original label is l), $x + \delta \in [0, 1]^m$ (input is within the valid pixel range) because the model has been normalized.

GNDF [19] is an improved version of DeepFool, the goal is to increase the robustness of DeepFool. They add Gaussian Noise to the f by $f_l^{'} = f_l(x_i) - f_{\hat{l}(x_i)}(x_0) + GN(x_i, l(x_i))$ which f is the score for every class computed by the DNN. In $f_l^{'} = f_l(x_i) - f_{\hat{l}(x_i)}(x_0) + GN(x_i, l(x_i))$, x_i is the original image input data or the output image data after previous iteration, $f_l(x_i)$ is the output of DNN, we often use the most high score as the finally output result, GN is the gaussian noise computed by x_i and $l(x_i)$. By adding Gaussian noise to the output of DNN, the direction of perturbation become single instead of multi-direction, and the robustness of the adversarial examples can be greatly improved.

Our approach DADF is also an optimization-based white-box attack method, by using the framework of DeepFool generate adversarial examples. DADF reduce the traversed target label set and towards to the most valuable labels to generate adversarial examples, so DADF is more effective than DeepFool.

V. Conclusions

In this paper, we analyze the label distributions of adversarial examples generated by DeepFool, then we propose the DADF method that can generate adversarial example more effective and faster. The DADF method has higher success rate than DeepFool, but it can save much time on generating adversarial examples. Experiments are conducted to validate the effectiveness of the proposed methods and explain why it can work in practice. The experimental results show that DeepFool algorithm has some bias for specific labels, which provides a possible way to optimizing adversarial attack algorithms and generating adversarial examples efficiently.

References

[1] Y. Jalalpour, L. Wang, R. Feng, and W. Feng, "Leveraging image processing techniques to thwart adversarial attacks in image classification," in 2019 IEEE International Symposium on Multimedia (ISM), pp. 184–1847, 2019.

[2] Q. Chen, J. Xu, and V. Koltun, "Fast image processing with fully-convolutional networks," in 2017 IEEE International Con- ference on Computer Vision (ICCV), pp. 2516–2525, 2017.

[3] Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL. Microsoft coco: Common objects in context". InEuropean conference on computer vision 2014 Sep 6 (pp. 740-755). Springer, Cham.

[4] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement, " 2018, arXiv:1804.02767. [Online]. Available: http://arxiv.org/abs/1804.02767.

[5] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn:towards real-time object detection with region proposal networks: Towards real-time object detection with region proposal net- works. In Advances in neural information processing sys- tems, 2015.

[6] Y.Zhang, K.Lee, and H.Lee. "Augmenting supervised neural networks with unsupervised objectives for large-scale image classification". CoRR, abs/1606.06582, 2016.

[7] C. Shi and C.-M. Pun, "Superpixel-based 3D deep neural networks for hyperspectral image classification," Pattern Recognit., vol. 74, pp. 600–616, Feb. 2018.

[8] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recog- nition, pp. 9185–9193, 2018.

[9] C. Szegedy et al., "Intriguing properties of neural networks," in Proc. ICLR, 2014.

[10] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv preprint arXiv:1412.6572, 2014.

[11] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2574–2582, 2016.

[12] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. (2017). "Ensemble adversarial training: Attacks and defenses," [Online]. Available: https://arxiv.org/abs/1705.07204.

[13] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang and R. Webb, "Learning from Simulated and Unsupervised Images through Adversarial Training," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 2242-2251.

[14] T. Miyato, S.-I. Maeda, M. Koyama, K. Nakae, and S. Ishii, "Distributional smoothing with virtual adversarial training," in Proc. ICLR, 2016, pp. 1–12.

[15] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," in 2017 IEEE International Conference on Computer Vision (ICCV), pp. 1378–1387, 2017.

[16] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in Proc. ACM Asia Conf. Comput. Commun. Security, 2017, pp. 506–519.

[17] S. Cheng, Y. Dong, T. Pang, H. Su, and J. Zhu. "Improving black-box adversarial attacks with a transfer-based prior,". In Advances in Neural Information Processing Systems, pages 10932–10942, 2019.

[18] T. Liiv and A. Strömberg, "Iterative, gradient-based adversarial attacks on neural network image classifiers," [Online] Available from: http://urn.kb.se/resolve?urn=urn:nbn: se:kth:diva- 255824, pp. 1–12, 2019.

[19] T. Xie and Y. Li, "Adding gaussian noise to deepfool for robustness based on perturbation directionality," Proceedings of the 2019 Australian Journal of Intelligent Information Processing Systems (ICONIP), vol. 16, no. 3, pp. 42–51, 2019.

Generating Adversarial Examples for Sentiment Classifier of Chinese Sentences

Yiting Zheng, Zhanqi Cui, Yue Xu, Haikuo Li, and Zhiwen Jiang
Computer School
Beijing Information Science and Technology University
Beijing, China
2017011321@mail.bistu.edu.cn, czq@bistu.edu.cn, xuyue@mail.bistu.edu.cn,
2017010588@mail.bistu.edu.cn, 2017010736@mail.bistu.edu.cn

***Abstract*—Studies have shown that deep learning models are vulnerable to adversarial examples, which cause incorrect predictions by adding imperceptible perturbations into normal inputs. The same characteristic goes for the sentiment orientation classification models. If the input text for the models contains perturbing information such as typos or special symbols, the inputs could misled the models. The adversarial examples reflect the diversity of text features, and the flaws of the sentiment orientation classification models can be found by adversarial attacks. The adversarial examples can be used to train the sentiment orientation classification models to improve the robustness of the model. This paper proposes to generate adversarial examples of Chinese sentences by replacing one of the characters in the word with similar Chinese characters in a black-box manner. The experimental results show that the adversarial examples generated by this method cost less and visually closer to the normal text.**

***Index Terms*—text classification, adversarial examples, robustness, sentiment tendency**

I. Introduction

With the rapid development of deep learning technology, deep neural networks (DNNs) are widely used in image recognition [1] [2], speech recognition [3], natural language processing [4] [5] [6] [7] and other fields. However, the security problems of deep learning system have been exposed gradually. Sgezedy et al. [8] proposed that deep learning models are vulnerable to be attacked by adversarial examples with small perturbations, it will lead the model gives a false output with high confidence. After that, Goodfellow et al. [9] explained the generation principle and application method of the adversarial examples. In their approach, adversarial examples are generated by adding some noises which are imperceptible to the human eyes as perturbations. The image recognition model will misclassify these adversarial examples, which seems no different from the original image, and the same goes for text and audio. The inability to effectively distinguish perturbation information makes the deep learning-based applications face huge security risks. The model is susceptible to perturbation information because the model is more sensitive to features with high generalization ability. By using the adversarial examples for training models can optimize the model and improve its robustness.

The security of sentiment analysis based on deep learning also faces huge threats. Sentiment classification technology is often used to extract people's attitudes or sentiments towards something by analyzing a large number of online comments. For example, through the analysis of sentiment classification technology, manufacturers can fully understand users' opinions on products and services. On the one hand, companies can use these suggestions to further improve product quality and adjust marketing strategies, on the other hand, governments can also use online comments on public events to broadly understand the public opinions as references for policymaking. In real life, these online comments are often mixed with perturbing information such as typos, acronyms, and network words. The sentiment orientation classification models are often misled due to this perturbing information. If an attacker exploits the weakness by adding some similar perturbations to malicious comments to generate a large number of adversarial examples and disseminate on the network, sentiment orientation classification models will be fooled to generate wrong results because the adversarial examples cannot be handled properly. For example, on an e-commerce platform, sentiment orientation classification models will be performed on customer consultation records in online customer service, so as to obtain users' opinions on products and provide guidelines for improving product quality. If the adversarial examples are disguised as normal data input, they will cause sentiment orientation classification model yields wrong improvement suggestions of users, which could lead to fails in competitions. Research on the generation of adversarial examples and analyze the security vulnerabilities of deep learning models can establish better defense mechanisms to prevent malicious adversarial attacks.

To generate adversarial examples for texts, there are some issues should be considered. First, the scenarios of adversarial example generation can be divided into white-box scenarios and black-box scenarios. In the white-box scenario, all information of the model can be obtained, including information

This work is supported in part by the Qin Xin Talents Cultivation Program of Beijing Information Science Technology University (No.QXTCP C201906) and 2020 Promote the Interior Development of University - Student Research and Training Project of Beijing Information Science and Technology University (5102010805).

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00013

such as network structures and weight parameters, and the adversarial examples are usually generated by gradient-based methods. In the black box scenario, the internal information of the model cannot be understood, and the adversarial examples can only be constructed by observing the external input and output. In most of the reality cases, the target models under attack are black box models. As a result, generating adversarial examples for black box models is more practical. Secondly, the text is discrete data, unlike continuous data, such as images and audios, the generation of text adversarial examples is more special and more challenging. Adding perturbation information to the text is easier to be perceived by human than images or audios. When generating text adversarial examples, the order of words and semantics are strictly limited. In order to make the text adversarial examples disguised as normal text, it is necessary to ensure the adversarial examples are understandable and seems similar to the meaning of the original text.

In this paper, we propose an approach for generating adversarial examples for sentiment orientation classification models of Chinese texts. At first, we use a scoring algorithm to measure the influence of each word in the input text on model in a black box manner. Then, we can get the score of each word. The word with higher is more sensitive the model. Finally, we select the word with the highest score to split into character sequences, and then replace one of the characters with a similar Chinese character and put it back into the original text. If the modified text fails to mislead the model, select the next word with the highest score for the same operation until the model is misclassified. The results show that the perturbation method of replacing similar words can generate adversarial examples that are more difficult for the human eyes to detect the traces of modification while keeping the original meaning of the sentence.

The main contribution of this paper can be summarized as follows:

- A novel approach for generating adversarial examples of Chinese texts, which replace a character that have the greatest influence on the sentiment orientation classification model with similar Chinese characters. The generated adversarial examples achieve a high attack success rate, at the same time, they only need a smaller modifications which are hard to be detected by human eyes.
- Experiments are performed on real datasets to compare the proposed approach with the homophone perturbation approach.

The rest of paper is organized as follows. Section 2 presents our method to generate adversarial examples. Section 3 carries out experiments and compares the methods between replacing characters with homophone and similar characters. Section 4 summarizes the related work of adversarial example generation. Section 5 concludes the paper.

II. Generating Adversarial Examples by Replacing Similar Chinese Characters

This paper generates adversarial examples in a black box manner, since we cannot obtain the relevant parameters of the target model. However, we can only obtain information by testing and observing the output of the model. It's obvious that the sentiment orientation classification model is more sensitive to some words, which affect sentimental tendency. Perturbing these words is easier to make the model get wrong outputs. In order to find these keywords that make the model sensitive to classification, we set up a method of searching for keywords. After the keywords that need to be perturbed are selected, they are perturbed by replacing the individual word of each word with similar Chinese characters. The adversarial examples generated in this way can keep original semantics and effectively simulate the features of text information.

A piece of original text data is inputted each time, and the importance score of each word in the sentence is calculated according to the scoring function after words segmentation. The words are sorted according to the score. Taking a word in the sorted order, split the word into character sequences. Only one of the characters is converted into a similar Chinese character at one time. The perturbed sentence is inputted into the target model and compare the new output label with the original label. If they are consistent, the output is a success adversarial example. If the output label and the original label are inconsistent, and the index does not exceed the word sequence, the next word is perturbed until the number of perturbations exceeds the number of words. If the number of perturbations exceeds the number of words, it is judged whether the editing distance is exceeded, if not, the next word are selected for perturbation, otherwise, the operation will be ended and fail to generate adversarial examples for the text.

A. Keywords Selection

Specifically, in order to make the generated adversarial examples more effective, it is necessary to select the words that have the greatest impact on the emotional probability score of the entire text for perturbation. Therefore, we set up a scoring function to evaluate which word are important for the target model, and in the next step we transform those words to form adversarial examples.

More detailed description is as follows: firstly, an original example is inputted and segmented to get a sequence of words. Then, one word is deleted from the sequence each time, the other words in the words sequence are spliced back to the original sentence, and then input it to the sentiment orientation classification model to record the prediction score, and calculate the difference with the original text. In equation (1), F is the function to get a prediction score of a text by the sentiment orientation classification model, w_i is the i^{th} word in the words sequence, *Score*(w_i) is the value of different model's prediction score between the original text and the text in which the i^{th} word is deleted.

$$Score(w_i) = |F(w_1, \ldots, w_(i-1), w_i, w_(i+1), \ldots, w_n) - F(w_1, \ldots, w_(i-1), w_(i+1), \ldots, w_n)| \quad (1)$$

B. Keywords Perturbation

After the keywords are selected, we design an transform algorithm to make the perturbed text visually more similar to the normal data. We choose to use similar words to replace individual words in the important words of the sentence for perturbation. These words are similar in structure to the original words in the original text. Some of them are just the difference between radicals. These small differences do not affect the reading to the human eyes. The perturbed words are more like some typos in form and conform to the text feature.

Using the perturbation method of changing a single character in a keyword's sequence into the similar Chinese character. The keywords sequence $W = <w_1, w_2, w_3, \ldots, w_n>$ is ordered by the scores. Suppose the edit distance is e, then the perturbation sequence is $<w_1, \ldots, w_e>(e<n)$, and each keyword in the perturbation sequence is divided into a character sequence $w_1 = <c_{1,1}, c_{1,2}, \ldots, c_{(1,n_1)}>, w_2 = <c_{2,1}, c_{2,2}, \ldots, c_{(2,n_2)}>, \cdots, w_e = <c_{e,1}, c_{e,2}, \ldots, c_{e,n_e}>$. Only one character is taken to be perturbated at one time according to the order of the character sequence. If a successful adversarial example is not generated, the edit distance is increased, and the next word in the perturbation sequence is taken to perturb. There are a total number of $(n_1 + n_1 * _2 + \cdots + n_1 * n_2 * \ldots * n_e)$ possibilities for perturbation.

Algorithm 1 Keywords Perturbation

Input: $X =< x_1, x_2, x_3, \ldots, x_n >$: original words sequence, e: edit distance

Output: X_s': adversarial examples set

```
Calculate the score of each word in X, and the sorting the words according to the scores to get a sequence of ordered keywords W :=< w_1, w_2, w_3, ..., w_n >
for i in range : 1 ~ e do
    w_i =< c_{i,1}, c_{i,2}, ... c_{i,n} > //the character sequence of w_i
    for c_{i,j} in w_i do
        c_{i,j}' := the similar Chinese character to c_{i,j}
        w_i' := Relpace c_{i,j} with c_{i,j}' in w_i
        X' :=Relpace x_i with w_i' in X
        if the classification of X' and X are different then
            add X' to X_s'
        end if
    end for
end for
return X'
```

III. Experiments and Evaluations

In this section, experiments are carried out on the black box model trained on a real Chinese text dataset. The experiment compares the effectiveness of the adversarial examples generated by the approach proposed in this paper and the homophone perturbation approach.

A. Experimental Design

The dataset used in the experiment is shown in Table 1 which is selected from a public Chinese sentiment analysis corpus [1]. 5 out of 10 fields are chose in the experiment, including hotel reviews from Ctrip.com and clothing, fruit, PDA, and shampoo from JD.com. , each field have 3000 records of texts, including 1500 positive reviews and 1500 negative reviews, and the average length of the texts are given in the last row of Table 1.

TABLE I
THE DATASET USED IN THE EXPERIMENTS

Datasets	Clothing	Fruit	PDA	Shampo	Hotel
Model	Sentiment tendency classification				
Size	3000	3000	3000	3000	3000
Types	2	2	2	2	2
Average text length	27	32	37	33	99

To verify the effectiveness of the proposed adversarial examples generated by replacing similar Chinese characters, we use the sentiment orientation classification model based on bi-LSTM and CNN in the Baidu sentiment analysis system Senta [2] as the target model, and input a piece of data to the target model , the target model will give the probability that the sentiment tendency is positive and negative (the sum of the two probabilities is 1), and then give the output classification. Through the scoring algorithm, we can calculate the importance of each word in a piece of text data depending on the output probability. In the experiment, the maximum edit distance is set to 3, and the adversarial examples are generated according to the perturbation method of replacing the similar Chinese characters, and the accuracy of the model is compared on the same datasets. The lower the model accuracy, the better the attack effect, and the generated adversarial examples can be more easily avoid the detection of sentiment tendency classification system and make it misclassify. At the same time, we also pay attention to the quality of the generated adversarial examples by checking whether it can retain the original semantics and be more like normal data.

B. Experimental Results

To compare the effectiveness of the proposed method with the perturbation method of replacing with homophone, experiments are carried out on the bi-LSTM and CNN models.

[1] ChineseNlpCorpus, https://github.com/SophonPlus/ChineseNlpCorpus/blob/master/datasets/online_shopping_10_cats/intro.ipynb.

[2] Senta, https://github.com/baidu/Senta.

TABLE II
The effectiveness of adversarial examples on the bi-LSTM model

Datasets	Original	Homophone		Similiar Chinese character	
	Acc(%)	Acc(%)	Decrease(%)	Acc(%)	Decrease(%)
clothing	99.40%	69.43%	29.97%	71.77%	27.63%
fruit	92.23%	61.13%	31.10%	66.53%	25.70%
pda	93.33%	65.47%	27.86%	68.67%	24.66%
shampoo	92.70%	62.63%	30.07%	65.67%	27.03%
hotel	88.70%	68.17%	20.53%	68.23%	20.47%

TABLE III
The effectiveness of adversarial examples on the CNN model

Datasets	Original	Homophone		Similiar Chinese character	
	Acc(%)	Acc(%)	Decrease(%)	Acc(%)	Decrease(%)
clothing	94.30%	72.23%	22.07%	73.37%	20.93%
fruit	92.07%	66.97%	25.10%	70.33%	21.74%
pda	92.70%	68.10%	24.60%	70.90%	21.80%
shampoo	92.63%	66.80%	25.83%	70.17%	22.46%
hotel	86.80%	70.30%	16.50%	69.43%	17.37%

As Table 2 and 3 show, we can find that the adversarial examples generated by the two perturbation methods can significantly reduce the accuracy of the model, and the perturbation method of replacing with similar Chinese character has a decrease of 25% on average for a bi-LSTM model and 21% on average for a CNN model, which are 3% and 1% less than the method of replacing with homophones.

It can be seen from Table 3 that in the case of generating adversarial examples based on the same data, the cost of perturbing by replacing similar Chinese characters is slightly greater than perturbing by replacing homophones.

C. Cost of Adversarial Examples

To compare the quality of the adversarial examples generated by the two different methods, we compare the adversarial examples that can be generated on the same data by the two methods. The perturbation cost in the two generation methods is shown in Table 4, e_{all} is the sum of edit distances, $\bar{e}$ as average edit distances, m is the modification on text, which is calculated as number of modified characters / total number of characters.

TABLE IV
Comparison of the costs of the two perturbation methods

Methods	e_{all}	$\bar{e}$	m
Homophone	7484	1.78	5.10%
Similar Chinese character	7631	1.81	5.20%

D. Text Semantic Deviation Measurements

To measure the semantic deviation of the adversarial example, we use the Word Mover's Distance (WMD) method to calculate the semantic similarity between the original example and the adversarial example. The WMD method is suitable for calculating the distance between texts. The larger the WMD distance, the lower the semantic similarity, and the greater the degree of semantic deviation . The smaller the WMD distance, the higher the semantic similarity, and the smaller the degree of semantic deviation. Examples of WMD calculation results are shown in Table 5.

The proportions of adversarial examples generated in two ways within different WMD distance intervals are shown in Table 6. It can be observed that the proportion of adversarial examples generated in different WMD distance intervals under the perturbation method of similar Chinese characters and homophones. They have similar results, the proportion of adversarial examples generated similar perturbations in the range of 0-0.2 is slightly higher than that of homophones.

E. Adversarial Examples Readability

We adopt a subjective evaluation method which can more accurately measure the readability of adversarial examples. 10 volunteers are recruited to evaluate the readability of adversarial examples. For every volunteer, we give a set of texts, each with 300 texts, including adversarial examples generated by the perturbation of homophones and similar words under different perturbation amplitudes, and then ask them questions: "Has this text been manually edited?", the time to observe each text and answer is controlled at about 3 seconds. The perception rate of adversarial examples with different edit distances (perception rate = number of adversarial example that has been manually recognized /total number of adversarial examples). The lower the human eyes perception rate, the better the readability of the text; on the contrary, the worse the readability of the text. Table 6 shows the relationship between the human eyes perception rate and the edit distance with different perturbation methods.

From Table 7, we can see that at the same editing distance, the human eyes perception rate of the adversarial examples generated by the way of replacing with similar Chinese characters with homophones is generally lower than that of homophone. This result indicates that the adversarial examples generated by the perturbation method of replacing with similar Chinese characters have lower human eyes perception rates. This results indicates that the adversarial examples generated by replacing with similar Chinese characters are easier to disguise as normal data, making it difficult for the human eyes to detect.

F. Adversarial Examples Generated

Some examples of the generated adversarial examples are shown in Table 8, '+' means positive classification, and '-' means negative classification. The first example is the adversarial examples generated by the perturbation method of replacing similar characters. The adversarial example is generated by changing the "最" character in the important word "最差" of the sentence. This makes the sentence, which is a negative comment, is misclassified by the model as a positive comment. The second example is generate by perturbing homophone replacements. The adversarial example is generated by changing the "很" and "位" in "很到位". The

TABLE V
Comparison of the costs of the two perturbation methods

Examples	WMD
收到货，外边很漂亮，切开就这样了，失望 收到货，外边很嘌亮，切开就適样了，失墅	0.24541437008186
没有送皮套，差评。 美有送批套，查评。	0.498499115107511
一用清扬,没有头皮屑。非常不错。 一用清扬,没有头皮屑。非常不措。	0.147250594264868
因为只住了一晚，所以没什么感觉，差不多四星吧。大堂的地砖很漂亮。房间小了一点。 音为只住了一晚，所以没身么感觉，差不多四星吧。大堂的地砖很票亮。房间小了一点。	0.123658136039004

TABLE VI
THE DATASET USED IN THE EXPERIMENTS

Methods	WMD					
	0-0.2	**0.2-0.4**	**0.4-0.6**	**0.6-0.8**	**0.8-1.0**	**>1.0**
Homophone	64.40%	28.92%	5.45%	1.09%	0.14%	0.00%
Similar Chinese character	64.85%	28.44%	5.43%	1.12%	0.16%	0.00%

TABLE VII
THE DATASET USED IN THE EXPERIMENTS

	e=1	**e=2**	**e=3**
Homophone	16.89%	32.40%	68.60%
Similar Chinese character	13.40%	20.34%	45.20%

sentences that should be a positive review were misclassified by the model as a negative review.

TABLE VIII
ADVERSARIAL EXAMPLES GENERATED

Original:	在京东买的最差的一次	-
Perturbed:	在京东买的最差的一次	+
Original:	当时有点坏了，不过服务很到位，解决了问题。	+
Perturbed:	当时有点坏了，不过服务狠到为，解决了问题。	-

IV. RELATED WORK

In recent years, more and more researches focused on the generation of adversarial examples. The generation of adversarial examples has made great progresses in the image processing field. The common generation methods are FGSM (fast gradient sign method) [9], DeepFool [10], etc. But the method of generating adversarial examples for images is not suitable for text data since the different data types. The image data is continuous, which can add noise to the pixels, and the text data is discrete and only by changing the words to generate adversarial examples. Generating adversarial examples by changing the content of the text and it is easy detectable by the human eyes, but we hope that the adversarial examples generated can be closer to the real data, so more ingenious perturbation methods are required.

Recent years, the research work on generating text adversarial examples has also made some progress, and these methods can be divided into two types according to scenes: white-box scene and black-box scene. If the target model is in a white-box scene, it means that we can know all the internal information of the model and we can use this information such as gradient to generate adversarial examples, while in the black-box scene, we can only use the input and output information of the model to generate adversarial examples.

A. White-box Attack Methods

Nowadays, there are many white-box attack methods. Liang et al. [11] proposed to find some hot words by the text gradient, and then perturb by inserting, modifying and removing words, but such perturbations makes the generated adversarial examples have a greater degree of semantic deviation. Papernot et al. [12] proposed to randomly change the words in the input text, replacing the words with the closest word vector as the perturbation. The generated adversarial examples in this way cannot maintain semantic similarity to the original samples. Ebrahimi et al. [13] used the method of synonym substitution, although the semantics were guaranteed, but the number of adversarial examples successfully generated within a limited edit distance was very rare. Jia et al. [14] proposed to insert attention in the text Disperse sentences to deceive the perturbation method of the reading system, but the perturbation method without inserting a short sentence is too obvious for short texts. For sentiment analysis systems, the input is generally short text type comments, which is not suitable for sentence-level perturbations.

B. Black-box Attack Methods

Although the attack methods used in the black-box scene are relatively few compared to the white-box, there have been some research results. Gao et al. [15] proposed the DeepWordBug algorithm, which uses a scoring algorithm to find important words in a sentence under black box scene, and then modify the word-level features of these important words,

such as swapping the alphabetical order of words, deleting letters, adding letters, replacing letters, etc. Such perturbation methods are only suitable for English texts, not for Chinese texts. We are more concerned about the application of Chinese texts. Wang et al. [16] proposed the WordHandling algorithm to deal with Chinese texts. The same way of scoring sentence words to find important words, using homophone replacement to perturb, but it is also prone to semantic deviations, and it is easier to be detected by the human eyes.

V. Conclusions

In this paper, we propose a new method to generate adversarial examples for sentiment orientation classification of Chinese sentences, experiments are conducted to compare it with the homophone method. The experiments results show that our perturbation method of replacing with similar Chinese characters can generate effective adversarial examples at a cost similar to the method of replacing with homophone, but the adversarial examples generated in our method are more difficult than homophone perturbation to be perceived by the human eyes and more inclined to be disguised as normal texts. In addition, the similar Chinese characters as a kind of visual modifications to text can be a new type of adversarial attack in Chinese texts. To improve robustness of the sentiment classifier models, the gaps in information processing between humans and models need to be considered to handle diverse types of data.

References

[1] Krizhevsky A, Sutskever I, Hinton G E, et al. "ImageNet Classification with Deep Convolutional Neural Networks," neural information processing systems, vol. 25, no. 2, 1097-1105, 2012.

[2] Taigman Y, Yang M, Ranzato M, et al. "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," computer vision and pattern recognition, pp.1701-1708 , 2014.

[3] Dahl G E, Yu D, Deng L, et al. "Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition," IEEE Trans, vol. 20, no. 1, 30-42 , 2012.

[4] Zhang X, Zhao J, Lecun Y, et al. "Character-level convolutional networks for text classification," neural information processing systems,pp.649-657, 2015.

[5] Pang B, Lee L, Vaithyanathan S, et al. "Thumbs up Sentiment Classification using Machine Learning Techniques," empirical methods in natural language processing, pp.79-86 , 2002.

[6] Sutskever I, Vinyals O, Le Q V, et al. "Sequence to Sequence Learning with Neural Networks," arXiv: Computation and Language, 2014.

[7] Maas A L, Daly R E, Pham P T, et al. "Learning Word Vectors for Sentiment Analysis," meeting of the association for computational linguistics, pp.142-150 , 2011.

[8] Szegedy C, Zaremba W, Sutskever I, et al. "Intriguing properties of neural networks," arXiv: Computer Vision and Pattern Recognition, 2013.

[9] Goodfellow I, Shlens J, Szegedy C, et al. "Explaining and Harnessing Adversarial Examples," arXiv: Machine Learning, 2014.

[10] Moosavidezfooli S, Fawzi A, Frossard P, et al. "DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks," computer vision and pattern recognition, pp.2574-2582 , 2016.

[11] Liang B, Li H, Su M, et al. "Deep Text Classification Can be Fooled," international joint conference on artificial intelligence, pp.4208-4215 , 2018.

[12] Papernot N, Mcdaniel P, Swami A, et al. "Crafting adversarial input sequences for recurrent neural networks," military communications conference, pp.49-54 , 2016.

[13] Ebrahimi J, Rao A, Lowd D, et al. "HotFlip: White-Box Adversarial Examples for Text Classification," meeting of the association for computational linguistics, pp.31-36 , 2018.

[14] Jia R, Liang P. "Adversarial Examples for Evaluating Reading Comprehension Systems," arXiv: Computation and Language, 2017.

[15] Gao J, Lanchantin J, Soffa M L, et al. "Black-Box Generation of Adversarial Text Sequences to Evade Deep Learning Classifiers," ieee symposium on security and privacy, pp.50-56 , 2018.

[16] Wang WQ, Wang R, Wang LN, Tang BX. "Adversarial examples generation approach for tendency classification on Chinese texts," Journal of Software, vol. 30, no. 8, pp.2415-2427, 2019.

A Controllable Hybrid Encryption Algorithm for Privacy Image

Yifeng Yin
School of Computer and Communication Engineering
Zhengzhou University of Light Industry
Zhengzhou, China
yinyifeng@zzuli.edu.cn

Chaofei Hu
School of Computer and Communication Engineering
Zhengzhou University of Light Industry
Zhengzhou, China
huchaofei_edu@163.com

Kunpeng Liu
School of Computer and Communication Engineering
Zhengzhou University of Light Industry
Zhengzhou, China
lkunpeng0716@163.com

Yong Gan
Zhengzhou Institute of Technology
Zhengzhou, China
ganyong@zzuli.edu.cn

***Abstract*—According to the security performance of image encryption, a hybrid image encryption algorithm is proposed. The algorithm can use different chaotic systems and pseudo-random generators to scramble and spread according to the needs of human control. The security of the algorithm is improved by dividing the plaintext into equal length sequence segments and encrypting them into inverse operations. Through the analysis of the sensitivity of the algorithm, the ability to resist differential attack and the encryption rate of the algorithm, it shows that the algorithm is safe and reliable.**

Keywords- Pseudorandom; Mixed encryption; Image encryption;

I. INTRODUCTION

With the rapid development of science and technology, information, multimedia and other information industries have developed one after another. With the coming of information age, information security has become the focus of people's attention. Therefore, it is very important to protect personal information and make it safe and reliable. Compared with the security of words, sounds and other information, image information contains more abundant information content and has higher confidentiality requirements. People hope that when transmitting information, people hope that their information content and transmission related information will not be intercepted or read by others, so as to ensure that their information content is complete and confidential[1].In communication information, image information is widely used, because image information contains a large amount of information, and can accurately describe information content and vivid image. In national defense, social, private and aerospace networks, image is the most prominent and frequent form of expression[2].Therefore, image information transmission is one of the common and convenient means in communication nowadays[3].So there are higher requirements for the security of image transmission.

This topic is aimed at the security and efficiency of image encryption. The security, encryption efficiency and storage space of image data are big problems. Compared with asymmetric encryption, symmetric encryption is more widely used in mobile terminals[4].In this paper, binary stream cipher is used for encryption, which has great advantages in efficiency and storage space. In addition, one-time encryption in stream cipher is used to ensure the security, so that the image can be transmitted safely and quickly. And because of the mixed encryption, in the aspect of scrambling and diffusion of image encryption, we can use the algorithm in line with its situation artificially. The overall algorithm of mixed encryption is controllable, and the algorithm security can be guaranteed, and the information security of image is reliable. This topic can make some private images can be transmitted safely and reliably, which is of great significance for image transmission involving privacy [5].

II. PRELIMINARIES

In recent years, with the increasing importance of image encryption, some theoretical knowledge in other fields has been applied to the research of image encryption technology [6-11],and many new image encryption methods[12] have been proposed. In 2011, Ebatatufar et al. Applied the theory of maximum stack tree[18] in image encryption algorithm[13-17], and achieved good scrambling effect. Moreover, the algorithm has high security, novel and unique ideas, but the computational complexity is very high. In 2012, Liu[19] et al. Applied DNA coding to image encryption, and used nucleotide coding to make pixel confusion, and proposed a novel confusion and diffusion method. In 2012, Abdullah[20] and others combined genetic algorithm with Logistic chaos to scramble the image, which greatly reduced the pixel correlation of the image. The research of image encryption technology applied in the computer has been mature, and some innovative image encryption algorithms have achieved good encryption effect, but these algorithms usually have high computational complexity[21], which is very important for the current society The actual value of is not high.

Therefore, some scholars combine chaotic system[22-27] with other encryption technologies to encrypt images, which

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00014

increases the complexity of image encryption and greatly improves its security. Since then, Zhao Feng and others put forward an image encryption algorithm[29] in reference[28], which is a combination of self coding, chaos and hyperchaos mapping. Compared with the above encryption algorithm, such algorithm[30,31] not only improves the security of image encryption, but also increases the encryption efficiency. In reference [32], an image encryption algorithm based on hyperchaos and bit substitution is proposed. The algorithm adopts two mapping methods, hyperhenon mapping and Kent mapping. In addition, a parameter is added to the encryption process. This parameter is closely related to the plaintext image, and the encryption effect and confidentiality are improved. In reference [33], an image encryption algorithm combining two-dimensional Kawakami and three-dimensional Lu is implemented. Firstly, the Kawakami hyperchaos sequence is generated by iteration for pixel replacement; then, the initial conditions and iterative parameters of Lu system are controlled by plaintext pixel value to generate the chaotic sequence for pixel replacement and diffusion.

In recent years, the researches on lightweight image encryption methods are few. And most of them use chaos encryption method to encrypt the image, which has low security. In this paper, we use the one-time encryption form of stream cipher and lightweight encryption algorithm to encrypt the image, which can encrypt the image quickly and safely

III. HYBRID ENCRYPTION ALGORITHM

A. *Encryption steps*

TABLE I SYMBOLS AND MEANINGS

Symbol	meaning
Se	Pseudorandom generator seed
Ks	Secret key
PRNG	Pseudorandom generator
M	Clear text picture
Gm	Plaintext sequence
Bs	Plaintext sequence
Mb	Chaotic sequence
Ss	Ciphertext sequence
Sm	Ciphertext matrix
S	Ciphertext picture
K1, K2, K3… Kn	One time key
C1, C2, C3… Cn	Plaintext sequence segment
S1, S2, S3… Sn	Ciphertext sequence segment

- After the image M is imported into Matlab and transformed into gray matrix, the gray matrix is transformed into binary sequence by Python as clear text sequence BS.
- As the initial value of chaos scrambling, the plaintext binary sequence BS is scrambled by logistic algorithm to generate chaos sequence MB.
- The generated chaotic sequence is divided into equal length sequence segments C1, C2, C3….Cn. The sequence segment is stored and transported through the chain list. The following one-time key and ciphertext sequence segment are stored and transported in the same way.

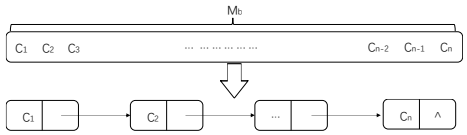


Figure 1 Sequence segment generation diagram

- Through pseudo-random generator, Select random seed Se and generate one-time key K1, K2, K3… Kn.
- One time key K1, K2, K3… Kn generated by pseudo-random generator The generated mixed sequence segments are encrypted one by one, and each one-time key only encrypts one chaotic sequence segment.
- Encrypt C1, C2, C3…Cn with one time key K1, K2, K3…. Kn Generate ciphertext sequence segment S1, S2, S3…Sn after encryption. Integrate S1, S2, S3…Sn into a whole ciphertext.

$$C_1 \oplus K_1 = S_1 \quad (1)$$

$$C_2 \oplus K_2 = S_2 \quad (2)$$

$$\cdots$$

$$C_n \oplus K_n = S_n \quad (3)$$

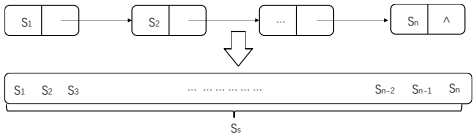


Figure 2 Sequence segment aggregation

- The ciphertext binary sequence is converted to ciphertext image S for transmission. The encryption process is shown in Figure 3:

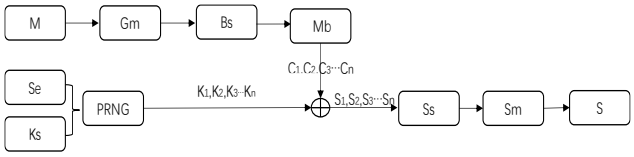


Figure 3 Encrption process

B. *Decryption steps*

After the ciphertext image s is imported into Matlab and transformed into gray matrix, the gray matrix is transformed

into ciphertext binary sequence Ss through Python. Then the ciphertext binary sequence is divided into equal length ciphertext sequence segments S1, S2, S3 … Sn. Then, the one-time key K1,K2,K3 …Kn. generated by the pseudo-random generator is used to decrypt the ciphertext sequence segment S1,S2,S3 …Sn to obtain the plaintext sequence segments C1, C2, C3 …Cn. The plaintext sequence segment is merged into the plaintext chaotic sequence Mb. The chaotic sequence Mb is reduced to plaintext sequence Bs by Logistic algorithm. Finally, the clear text sequence Bs is transformed into the clear text picture. This is the whole decryption process.

IV. SIMULATION RESULT

The simulation of this experiment is carried out through MATLAB.Fig.4 shows the result of the simulation experiment with Lena image as the material.Fig.4(a) shows the original image before encryption,Fig.4(b) shows the encrypted ciphertext image, and Fig.4(c) shows the image after correct decryption. Fig.5(a) is the pixel distribution histogram of the plaintext image, and Fig.5(b) is the pixel histogram of the encrypted image. Compared with Fig.5(a),it can be seen that the difference between the pixel distribution after encryption and that before encryption is large, and the image information is well hidden. Compared with Fig.4(a), Fig.4(c), Fig.5(a) and Fig.5(c), it can be seen that the decrypted image is exactly the same as the image histogram, and there is no loss of image information in the process of encryption and decryption.

Figure 4 Plaintext image, (b) Ciphertext image, (c) Decrypt image

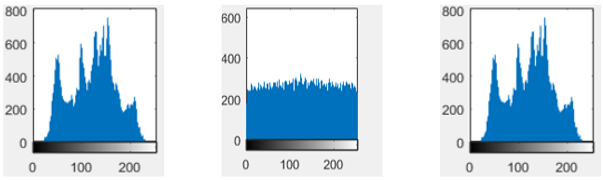


Figure 5 Plaintext histogram, (b) Ciphertext histogram, (c) Decrypt histogram

V. ALGORITHM ANALYSIS

A. *Sensitivity analysis*

The iterative expression of logistic scrambling is as shown in formula (1), but it can be seen from experiments that not all μ can iterate out of chaos. Through experiments, we know that when 0 < x_ 0 < 1 and 3.5699456 When < μ < = 4, the logistic function works in chaos. And when the value of μ is close to 4, the effect of chaos is better.

$$x_{n+1} = \mu x_n (1 - x_n) \tag{4}$$

Because for different μ values, logistic will produce different chaotic states When < μ < = 4, the chaotic state is more obvious. And when 3.5699456 When < μ < = 4, the whole chaotic system will become very sensitive to the change of initial value. The initial value is only slightly different, and the result after iterative chaos is also very different. The following table shows the selected μ = 4, the initial values are 0.4500000 and 0.4500001 after different iterations and the difference between them. The experimental results are shown in the Table II:

TABLE II SENSITIVITY TEST RESULTS

	0.4500000	0.4500001	Difference between the two
1 time	0.990000000	0.990000040	0.000000040
5 times	0.998983856	0.998984061	0.000000205
10 times	0.726299601	0.726207825	0.000091776
15 times	0.889330819	0.887256022	0.002074797
20 times	0.979601832	0.998562280	0.018960448
30 times	0.616660231	0.823703566	0.207043335
40 times	0.862469682	0.458958300	0.403511382
50 times	0.241996508	0.890376834	0.648380326

It can be seen from the table that when μ = 4 and the initial value is only 0.0000001, after many iterations, the results are very different, so the encryption method has the initial value sensitivity and unpredictability.

B. *Anti differential attack analysis*

A secure image encryption algorithm should have good anti differential attack ability. Generally speaking, for image encryption, two standards are usually used to measure the anti differential attack ability of this algorithm on image. The two standards are number of pixels change rate (NPCR) and unified average changing intensity (UACI). NPCR and UACI are calculated as follows:

$$NPCR = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\% \tag{5}$$

$$UACI = \frac{1}{M \times N}\left(\sum_{i,j} \frac{\left| C_1(i,j) - C_2(i,j) \right|}{255} \right) \times 100\% \tag{6}$$

$$D(i,j) = \begin{cases} 1, C_1(i,j) \neq C_2(i,j) \\ 0, C_1(i,j) = C_2(i,j) \end{cases} \tag{7}$$

C1, C2 are two gray-scale images of (M,N) size

Table III shows the experimental analysis results. The parameters used for the NPCR and UACI values are the ciphertext image encrypted by the original plaintext and the image encrypted by the minimum changed plaintext. After many experiments, the maximum, minimum and average values of the NPCR and UACI results are calculated and compared with the theoretical values, Compared with literature [34~37], the algorithm in this paper has the same change of plaintext, and its sensitivity is stronger, so its anti differential attack ability is stronger than literature [34~37]. The experimental results are shown in the Table III:

TABLE III ANTI DIFFERENTIAL ATTACK TEST RESULTS

Algorithm	Max		Min		Average		Theoretical	
	NPCR	UACI	NPCR	UACI	NPCR	UACI	NPCR	UACI
This Paper	0.9963	0.3356	0.9958	0.3339	0.9961	0.3346	0.9961	0.3346
Paper [34]	0.9964	0.3358	0.9958	0.3335	0.9961	0.3347	0.9961	0.3346
Paper [36]	0.9973	0.3355	0.9954	0.3338	0.9962	0.3345	0.9961	0.3346
Paper [37]	0.9968	0.3367	0.9955	0.3331	0.9961	0.3346	0.9961	0.3346

TABLE IV ENCRYPTION TIME

Image Size	This Paper	Paper[34]	Paper[35]	Paper[36]	Paper[37]	Paper[38]
256x256	0.16	0.15	0.12	0.46	0.18	1.05
512x512	0.31	0.32	0.54	1.26	0.49	3.68
1024x1024	0.89	0.95	2.78	3.88	1.52	14.04

C. Algorithm encryption rate

In addition to security, the quality of an algorithm is also closely related to the encryption speed of the algorithm. If the security performance of an algorithm is very high, but the encryption speed of this algorithm is particularly slow, in today's multimedia big data era, the actual meaning of this algorithm appears to be relatively low. Although the security of the algorithm is higher, there are fewer places that can be used in practical applications. The encryption object of the encryption algorithm in this paper is images, and the amount of data is much larger than that of text. Therefore, the encryption speed of this algorithm is more important than traditional encryption methods.

The encryption algorithm in this paper uses binary stream block encryption, which is faster than the reference [34 ~ 37] in terms of encryption speed. According to the data analysis, the bigger the image is, the more obvious the advantage of encryption speed is. In the multimedia era with more and more data, the encryption algorithm in this paper is more suitable for this era. The encryption speed of different paper for different size images is shown in Table IV:

VI. CONCLUSION

This paper presents a hybrid encryption algorithm, which combines scrambling algorithm with binary stream block encryption. Firstly, the binary sequence is scrambled into chaotic and disordered sequence by logistic algorithm, and then the chaotic and disordered sequence is divided into sequence segments of equal length, and then each segment is encrypted by one-time key generated by pseudo-random generator. Because the use of scrambling and diffusion of two encryption methods to encrypt the sequence, high security performance, and through binary stream block encryption also improves the speed of encryption, for big data encryption objects, the encryption algorithm in this paper has a certain practical value. And the hybrid encryption algorithm can be adjusted according to different scenarios, which is a controllable encryption algorithm. Experimental data show that the key sensitivity of this algorithm is relatively high, and the ability to resist differential attack is also relatively strong. In small and medium-sized images, the encryption speed of this paper is general, and on large-scale images, the encryption speed of this paper is obvious. All in all, the security of the algorithm is also very high.

ACKNOWLEDGMENT

This research is supported by National Natural Science Foundation of China under Grant No.61572445, U1804263 and 61272038. Key foundation of Science and Technology Development of Henan Province No.142102210081.

REFERENCES

[1] Niu Xiaoyu. Research on image encryption algorithm based on multidimensional chaotic system [J]. Science and technology innovation, 2019 (19).

[2] Lakshmi C, Thenmozhi K, Rayappan J B B, et al. Hopfield attractor-trusted neural network: an attack-resistant image encryption[J]. Neural Computing and Applications, 2019: 1-13.

[3] Sun Hepeng, Zhang Xiaoqiang. Multiple image encryption algorithm based on DNA coding [J]. Computer engineering and design, 2018, 39 (10): 58-62 + 107.

[4] M.Emadaldeen, Z.Saadi. Color Image Encryption Depend on DNA Computing. Diyala Journal For Pure Sciences, 2017, 1(13):74-94.

[5] Z.Yong, H.WenGang. A Fast Image Encryption Algorithm Using Plaintext-related Conffusion. Proceeding of 2016 IEEE Information Technology, Network, Electronic and Automation Control Conference and, 2106, 23:293-297.

[6] Liu Jingyi. Research on digital image encryption algorithm based on chaos system [D]. Hubei University for nationalities, 2019.

[7] Wang Xingyuan,Zhang Junjian,Cao Guanghui. An image encryption algorithm based on ZigZag transform and LL compound chaotic system[J]. Optics and Laser Technology,2019,119.

[8] Li Jing, Xiang Fei, Zhang Junpeng. A scheme of digital image information encryption based on chaos [J]. Electronic design engineering, 2019,27 (12): 84-88.

[9] Qiu Jin, Wang Ping, Xiao Di, Liao Xiaofeng. Pseudorandom sequence generator based on chaotic mapping [J]. Computer science, 2011,38 (10): 81-83.

[10] Wu Xiaogang. Construction and application of pseudo-random sequence generator in stream cipher [J]. Journal of Xingyi Normal University for nationalities, 2011 (02): 105-107 + 115.

[11] Li Chunhu. Research on Key Technologies of image encryption based on chaos [D]. University of Electronic Science and technology, 2018.

[12] H.XiaoLing, Y.GuoDong. An image encryption algorithm based on hyper-chaos and DNA sequence. Multimedia Tools and Applications. 2014,72(1):57-70.

[13] H.Ting, L.Ye, G.Hua, G.Feng, et al. Chaotic image cryptosystem using DNA deletion and DNA insertion. Signal Processing, 2017, 134 : 234-243.

[14] W.XiangJun, W.KunShu, W.XingYuan, et al. Lossless chaotic image cryptosystem based on DNA encryption and entropy. Nonlinear Dyn. 2017, 90:855-875.

[15] K.C. Jithin,Syam Sankar. Colour image encryption algorithm combining, Arnold map, DNA sequence operation, and a Mandelbrot set[J]. Journal of Information Security and Applications,2020,50.

[16] R. Vidhya,M. Brindha,N. Ammasai Gounden. A secure image encryption algorithm based on a Parametric Switching Chaotic System[J]. Chinese Journal of Physics,2019

[17] Islam T. Almalkawi,Rami Halloush,Ayoub Alsarhan,Ahmed Al-Dubai,Jamal N. Al-karaki. A lightweight and efficient digital image encryption using hybrid chaotic systems for wireless network applications[J]. Journal of Information Security and Applications,2019,49.

[18] Enayatifar R. Image encryption via logistic map function and heap tree[J]. International Journal of Physical Sciences, 2011, 6(2): 221-228.

[19] Hermassi H, Belazi A, Rhouma R, et al. Security analysis of an image encryption algorithm based on a DNA addition combining with chaotic maps[J]. Multimedia tools and applications, 2014, 72(3): 2211-2224.

[20] Abdullah A H, Enayatifar R, Lee M. A hybrid genetic algorithm and chaotic function model for image encryption[J]. AEU-International Journal of Electronics and Communications, 2012, 66(10): 806-816.

[21] Xingyuan Wang,Yu Wang,Salahuddin Unar,Mingxu Wang,Wang Shibing. A privacy encryption algorithm based on an improved chaotic system[J]. Optics and Lasers in Engineering,2019,122.

[22] Koearev,Jakimovski G. Chaos and cryptography:From chaotic maps to encryption algorithms[J].IEEE Transactions on Circuits and Systems-I. 2001, 48(2):163-169.

[23] Hu G.S. Hyperchaos of higher order and its circuit implementation[J].International journal of Circuit Theory and Applications, 2011, 39(1): 79-89.

[24] Swathy P.S.,Thamilmaran K. Hyperchaos in SC-CNN based modified canonical Chua's circuit[J].Nonlinear dynamics,2014, 78(4):2639-2650.

[25] Mahmoud G.M.,Al-Kashif M.A.,Farghaly A.A.Chaotic and hyperchaotic attractors of a complex nonlinear system[J].Journal of physics A-mathematical and theoretical, 2008, 41(5): 055104.

[26] Liu S.T.,Liu P. Adaptive anti-synchronization of chaotic complex nonlinear systems with unknown parameters[J].Nonlinear Analysis:Real World Applications, 2011, 12(6): 3046-3055.

[27] Wu G.C., Baleanu D. Chaos synchronization of the discrete fractional logistic map[J].Signal Processing, 2014, 102: 96-99.

[28] Zhao Feng, Wu CHENGMAO. Image encryption algorithm combining self coding and hyperchaotic mapping [J]. Journal of computer aided design and graphics, 2016,28 (1): 119-128.

[29] Kanso A. Self-shrinking chaotic stream ciphers[J]. Communications in nonlinear science and numerical simulation,2011,16(2):822-836.

[30] Lu Xiang, Lu Yuefeng, Guo Yuhui, Lou Mengjia, Liu Shuang, Qiang Yijia, Deng Yuting, fu na. An image encryption algorithm based on pseudo features [J]. Journal of Zhejiang Normal University (NATURAL SCIENCE EDITION), 2019,42 (04): 393-399.

[31] Akhshani A , Akhavan A , Mobaraki A , Lim S.C., Hassan Z. Pseudo random number generator based on quantum chaotic map. Communications in nonlinear science and numerical simulation, 2014, 19(1): 101-111.

[32] Xie Guobo, Wang Tian. A new hyperchaotic image encryption algorithm based on bit scrambling [J]. Microelectronics and computer, 2016,33 (7): 28-32.

[33] Wu Yifeng, Gou Xinke. Image encryption algorithm based on the combination of two-dimensional hyperchaos and three-dimensional chaos [J]. Electro optic and control, 2018,25 (11): 46-51.

[34] Zhao Hongxiang, Xie shucui, Zhang Jianzhong, Wu Tong. A fast image encryption algorithm based on improved Henonmapping[J/OL].Computer application research: 1-6 [2020-05-24].

[35] Wang Xingyuan, Liu Lintao, Zhang Yingqian. A novel chaotic block image encryption algorithm based on dynamic random growth technique [J]. Optics and Lasers in Engineering, 2015, 66 (3): 10-18.

[36] Chai Xiuli, Gan Zhihua, Zhang Miaohui. A fast chaos-based image encryption scheme with a novel plain image-related swapping block permutation and block diffusion [J]. Multimedia Tools and Applications, 2017, 76 (14): 15561-15585.

[37] Wang Mingxu, Wang Xingyuan, Zhang Yingqian, et al. A novel chaotic encryption scheme based on image segmentation and multiple diffusion models [J]. Optics and Laser Technology, 2018, 108 (12): 558-573.

[38] Rehman A, Xiao Di, Kulsoom A, et al. Block mode image encryption technique using two-fold operations based on chaos MD5 and DNA rules [J]. Multimedia Tools and Applications, 2019, 78 (7): 9355-9382

Dynamic Workflow Scheduling based on Autonomic Fault-Tolerant Scheme Selection in Uncertain Cloud Environment

Chenyang Zhao
Key Laboratory of Grain Information Processing and Control (Henan University of Technology)
Ministry of Education
Zhengzhou, China
zhaochy2005@163.com

Junling Wang
School of Science
Henan University of Technology
Zhengzhou, China
wangjl2013@163.com

***Abstract*—Most of the existing methods cannot deal with the problem of fault-tolerant workflow scheduling in uncertain environment. Therefore, the paper proposes a dynamic workflow scheduling based on autonomic fault-tolerant scheme selection in uncertain cloud environment. In order to make full use of various cloud resources, the tasks that constitute workflow are divided into various types. And a computing capacity fluctuation model based on multiple states and jump points is established to describe the instability of cloud resource. Moreover, an extended fault-tolerant model is built to deal with various computation and transmission faults. Based on the two models and task classification, tasks are firstly assigned statically to appropriate virtual machines, then the fault-tolerant strategies and the actual execution for tasks are automatically adjusted in a dynamic manner. Experiments show that the proposed method can perform effectively in uncertain cloud environment while ensuring fault tolerance.**

Keywords-workflow scheduling; uncertain cloud; fault tolerance; fluctuation; task classification

I. INTRODUCTION

Workflow composed of a set of dependent tasks is a new data analysis and scientific computing mode, which is widely used in biology, physics, astronomy, geography and other fields [1]. Because the execution of workflow needs large-scale data transmission and intensive scientific computing, the powerful processing ability of running environment is necessary for workflow. Cloud systems provide users with computing resources, storage resources, network resources and other resources through virtualization technology, which makes it possible for the successful execution of workflow [2]. Besides, cloud systems also provide the resource in the pay-as-you-go model [3], which makes more and more users willing to submit workflow to cloud systems for execution.

Although cloud systems provide a running environment, there are still several problems in the execution of workflow in cloud systems due to the complexity of workflow and the uncertainty of could environment. Specifically, they are manifested in the following three aspects. Firstly, workflow usually contains various types of tasks, such as computing-intensive task and data-intensive task. Most of existing researches mainly focus on the scheduling of compute-intensive tasks, while little attention has been paid to the scheduling of hybrid tasks. Secondly, due to the uncertainty of cloud environment, the occurrence of various faults is inevitable [4]. When tasks cannot be executed due to faults, it is very important to adopt an effective fault-tolerant model to ensure the execution of the tasks. These faults are usually caused by computing resource or data transmission. However, the current fault-tolerant researches mainly focus on the faults from computing resources, while the researches on the faults from data transmission is rare. Thirdly, the uncertainty of cloud environment mainly includes the computing capability fluctuation of virtual machine and various types of faults, which will affect the execution of workflow. It is necessary for workflow scheduling to make adaptive response to the uncertainty. However, the current researches in this topic are not enough.

To solve above problems, a workflow scheduling based on autonomic fault-tolerant scheme selection (WSAFSS) in uncertain cloud environment is proposed. In comparison with the related works, the main contributions of the paper can be summarized as follows. First, both computing-intensive tasks and data-intensive tasks are considered, which is beneficial to use the advantages of various cloud resources. Second, a computing capacity fluctuation model based on multiple states and jump points is established to describe the unstable computing capability of cloud resource. Third, an extended fault-tolerant model including virtual machine faults and data transmission faults is built, and different fault-tolerant strategies are formulated according to task natures and fault types. In final, in uncertain cloud environment, a WSAFSS scheduling algorithm consisted of static scheduling and dynamic scheduling is proposed, which can adaptively adjust fault-tolerant strategies and execution for tasks.

The remainder of the paper is organized as follows. Section 2 reviews the related works. The computing capacity fluctuation model is defined in section 3. In section 4, the extended fault-tolerant model is described. The proposed WSAFSS method is presented in section 5. In section 6, some experiments and analyses are implemented. In the final section, the paper is concluded.

II. RELATED WORK

At present, researchers usually take one or more aspects as optimization objectives of workflow scheduling, including

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00015

finish time, cost, energy and workload, and put forward different workflow scheduling methods combined with various fault-tolerant strategies. Masdari et al. made a comprehensive analysis and summary of the current mainstream workflow scheduling methods in [5]. According to the types of workflow and optimization objectives, the workflow scheduling methods were divided into metaheuristic-based scheduling, heuristic-based scheduling and hybrid scheduling. In [6], Yao et al. designed a workflow scheduling algorithm in the cloud, which simulated the working principle of human immune system to solve the problem of task failure. Ding et al. proposed a fault-tolerant scheduling method for workflow based on primary and backup model (PB model) in [7], which considered the constraints between tasks with dependency in workflow and the elastic characteristics of cloud resources. Besides, the utilization of system resources was further improved through the migration strategy of virtual machine. In [8], Han et al. proposed a fault-tolerant scheduling method for real-time tasks based on CPB model, which uses the strategy of PB model and checkpoint to solve task failure. In [9], Vinay et al. proposed a workflow scheduling method based on cost and fault tolerance, which employed bidding strategy to reduce cost, and used replication and checkpoint for key tasks to reduce failure rate and cost. Lee et al. proposed an effective fault-tolerant task scheduling method in mobile cloud environment by combining checkpoint and replication in [10]. Wu et al. proposed a dynamic fault-tolerant workflow scheduling with hybrid spatial-temporal and re-execution in clouds, which considered the urgency and budget of tasks simultaneously in [11]. Zhang et al. proposed a workflow scheduling which ensured the deadline constraint through a distance algorithm [12]. The above researches mainly focus on the scheduling of computing-intensive tasks, while the scheduling of multi-type tasks is less studied. In addition, it is assumed that the computing capacity of cloud resource during the scheduling is stable. In fact, due to the complexity and variability of cloud environment, the computing capacity of cloud resource is uncertain.

For the uncertain cloud environment, some researchers had made some attempts. In [13], Chen et al. used interval number theory to describe the uncertainty of cloud environment, and proposed a scheduling method for real-time tasks in the uncertain cloud environment, in which task accomplish, resource utilization and energy consumption were well balanced. In [14], Yan et al. proposed a dynamic, flexible and fault-tolerant workflow scheduling method when the task execution time is uncertain. In this method, the task completion time was limited to a certain range, which represents the uncertainty of task execution time. Based on the assumption, two fault-tolerant strategies of resubmission and replication were adopted to design the workflow scheduling. In [15], Malakoutifar et al. firstly established the performance fluctuation model of computing resources by using Markov chain, and then proposed a fault-tolerant workflow scheduling. In summary, although these researches on workflow scheduling have achieved some good results, the models of uncertain cloud environment are not perfect and detailed enough, and the possibility of data transmission failure is not concerned in these fault-tolerant models.

III. CLASSIFICATION OF TASK AND VIRTUAL MACHINE

The types of tasks which make up the workflow may be different, and different task types have different resource requirements for the virtual machines. In order to reduce task finish time and make full use of virtual machines, both tasks and virtual machines are firstly classified in the WSAFSS method. According to the computing time and data transmission time of tasks, the tasks are classified in [8], but the differences of time duration between tasks belonging to the same task type are not considered. Therefore, a more reasonable classification method for tasks and virtual machines is proposed in the paper.

A task usually consists of two parts: data computing and data transmission. For the entire task set, given fixed computing and data transmission capabilities, the duration of time is divided into four time periods, namely four levels, which are used to measure the different durations of computing time or data transmission time required by the task. The higher the level of time is, the longer the duration is. Assume that the computing time and data transmission time of the task t_i belong to level $cl(t_i)$ and level $tl(t_i)$ respectively. According to the following rules, the task t_i may be divided into six groups: 1) single-strong computing-intensive task (SS-CT): the task meets the conditions $cl(t_i) > tl(t_i)$ and $tl(t_i) \le 2$. Its finish time mainly depends on the computing capability of virtual machine; 2) double-strong computing-intensive task (DS-CT): the task meets the conditions $cl(t_i) > tl(t_i)$ and $tl(t_i) > 2$. Its finish time mainly depends on the computing capability of virtual machine, but the requirement of data transmission capacity is stronger than SS-CT; 3) single-strong data-intensive task (SS-DT): the task meets the conditions $cl(t_i) < tl(t_i)$ and $cl(t_i) \le 2$. Its finish time mainly depends on the data transmission capability of virtual machine; 4) double-strong data-intensive task (DS-DT): the task meets the conditions $cl(t_i) < tl(t_i)$ and $cl(t_i) > 2$. Its finish time mainly depends on the data transmission capability of virtual machine, but the requirement of the computing capability is stronger than SS-DT; 5) single-strong balanced task (SS-BT): the task meets the conditions $cl(t_i) = tl(t_i)$ and $cl(t_i) \le 2$. Its finish time depends on the entire computing and data transmission capacity of virtual machine; 6) double-strong balanced task (DS-BT): the task meets the conditions $cl(t_i) = tl(t_i)$ and $cl(t_i) > 2$. Its finish time depends on the entire computing and data transmission capacity of the virtual machine, but the requirement is stronger than SS-BT.

Accordingly, according to the computing and data transmission capacity of virtual machine, the virtual machines are divided into six categories: single-strong or double-strong computing virtual machine, single-strong or double-strong transmission virtual machine and single-strong or double-strong balanced virtual machine.

IV. THE COMPUTING CAPACITY FLUCTUATION MODEL

Because of the uncertainty in cloud environment, the computing capacity of cloud resource is usually unstable, which is manifested as fluctuation with time. Referring to [15], a novel and extended computing capacity fluctuation model of virtual machine based on multiple states and jump points is proposed in the paper, as shown in Fig. 1. In the model, the computing capacity of virtual machine is firstly divided into several state types, and then the actual computing capacity fluctuation of virtual machine is simulated as accurately as possible through multiple state types and the transformation of these state types.

Let $ST = \{ST_1, ST_2, \cdots, ST_K\}$ be the K state types, where ST_i represents the ith state type of computing capacity of a given virtual machine v_j. At any moment, the computing capacity state s_k of v_j belongs to one of K state types. For the state type ST_i, the computing capacity of v_j is relatively stable, usually fluctuating around a basic value ε_i with a small margin. The fluctuation function is recorded as $F_i(t)$, for example, the function used in this model is $F_i(t)=\varepsilon_i+\sigma_i \sin(t)$, where σ_i is the parameter controlling small fluctuation. The model also assumes that there is a jump point between two adjacent state types, which represents the sudden change of computing capacity of v_j. The relations between these basic values are defined as $\varepsilon_i=\varepsilon_1[1-(i-1)\times\delta]$, $1 \le i \le K$, where δ is the parameter controlling the jumping range of computing capacity.

During the running of virtual machine, the state types of its computing capacity states may change constantly, and the state may be transformed from one state type to another. The transformation process is abstracted as Markov process in the model. Assume that the current computing capacity state s_k of v_j belongs to ST_i. When $1<i<K$, the state type of next state may be transformed to ST_{i+1} with probability $p_{i,i+1}$, or to ST_{i-1} with probability $p_{i,i-1}$, and $p_{i,i+1}+p_{i,i-1}=1$. When $i=1$ (or $i=K$), the next state may be remained in the original state type with probability $p_{1,1}$ (or $p_{K,K}$), or be transformed to ST_2 (or ST_1) with probability $p_{1,2}$ (or $p_{K,K-1}$), and $p_{1,1}+p_{1,2}=1$ (or $p_{K,K}+p_{K,K-1}=1$). The transformation probability matrix between state types is recorded as $P=\{P_{i,j} \mid 1 \le i \le K, 1 \le j \le K\}$.

As shown in Fig. 1, the sequence of computing capacity states of virtual machine is recorded as $s=\{s_1, s_2, \cdots, s_n, \cdots\}$. Let $T_i = t_{i+1} - t_i$ represent the duration of the state s_i, then the state duration sequence is recorded as $T=\{T_1, T_2, \cdots, T_n, \cdots\}$. Assume that T_i is independent of each other and follows the exponential distribution $F(t)=1-e^{-\lambda t}$ $(t \ge 0)$ with parameter λ, the expected duration of T_i is $E(t)=1/\lambda$. Let $P(n)=P^n$, according to the characteristics of the matrix P, there is a positive integer l, when $l=K-1$, for any i and j, $P_{i,j}(K-1)>0$. According to the properties of Markov chain,

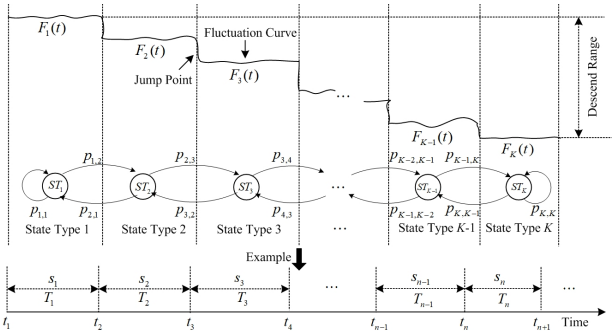


Figure 1. Computing capacity fluctuation model of virtual machine

there is a limit distribution, that is, when $n \to \infty$, matrix P converges to matrix P', as shown in (1). In the matrix P', for any i, $1 \le i \le K$, then $\sum_{j=1}^{K} P'_{i,j}=1$, and for any j, $1 \le j \le K$, then $P'_{i,j}=q_j, 1 \le i \le K$.

$$\lim_{n\to\infty} P(n) = \lim_{n\to\infty} P^n = P' \tag{1}$$

Let the probability distribution of state types at the initial computing capacity state of virtual machine be $\alpha^T=(\alpha_1, \alpha_2, \cdots, \alpha_K)$, where α_i is the probability of the initial computing capacity state belonging to the state type ST_i. According to (1), when $n \to \infty$, the probability distribution tends to be stable, and is independent of the initial probability distribution, as shown in (2).

$$\lim_{n\to\infty} \alpha^T P^n = \alpha^T P' = q^T \tag{2}$$

From the above analysis, the probability distribution of computing capacity state of v_j belonging to K state type is q, and the expected duration of each state is $E(t)$, then the expected computing capacity $c_i(v_j)$ of v_j at the state type ST_i is shown as (3), and the total expected computing capacity $c_e(v_j)$ of v_j is shown as (4).

$$c_i(v_j) = \int_0^{E(t)} F_i(t)dt \Big/ E(t) = \lambda \int_0^{1/\lambda} (\varepsilon_i + \sigma_i \sin(t))dt \tag{3}$$

$$c_e(v_j) = \sum_{i=1}^{K} q_i c_i(v_j) \tag{4}$$

V. THE EXTENDED FAULT-TOLERANT MODEL

In order to solve the failure of task execution caused by the uncertainty in the cloud environment, different fault tolerance strategies should be adopted according to different situations. At present, most researches mainly focus on host or virtual machine fault, but seldom on data transmission fault. Therefore, an extended fault-tolerant model including virtual machine fault and data transmission fault is proposed in the paper. These faults are divided into temporary faults and permanent faults. And assume that only one host or virtual machine fails, or the data transmission connected to

one host or virtual machine fails at any moment. For multiple and complicated faults, grouping methods can be used [7,14].

There are four fault-tolerant strategies used in the model, which are task breakpoint execution (TBE), task re-execution (TRE), primary and backup (PB), and connection bridge transmission (CBT). TBE represents that when the fault is temporary, the task continues to execute on the original virtual machine of the original host after the fault recovers automatically. TRE represents that no matter whether the fault is temporary or permanent, the task is re-executed on a virtual machine of another host. PB represents that the primary and backup versions of the task are executed on different virtual machines of different hosts. CBT refers to that when the data transmission between the sender virtual machine and the receiver virtual machine fails to transmit data, the sender first sends the data to a specific bridge virtual machine agent, and then the agent forwards the data to the receiver. The selection of fault-tolerant strategies is determined by workflow scheduling method according to task type, task nature and fault type.

VI. The Proposed Method

The proposed WSAFSS method mainly focus on four aspects as follows. First, assigning task to appropriate virtual machine according to task type. Second, considering the importance of task in workflow. Third, paying attention to the computing capacity fluctuation of virtual machine. In final, due to the computing capacity fluctuation of virtual machine and the fault occurrence, task nature may change, so the fault-tolerant strategy for task should be adjusted adaptively during workflow execution. Specifically, the proposed WSAFSS method includes the static scheduling and the dynamic scheduling.

A. The Static Scheduling

In the static scheduling, tasks in workflow are firstly assigned to the appropriate virtual machines according to task types, and then the nature of each task is determined, that is, whether the task is critical or non-critical task. At present, most of the scheduling methods focus on the scheduling of computing-intensive tasks under the stable cloud environment. Different from these existing methods, the proposed method not only considers the scheduling of tasks with various types, but also focuses on the impact of the computing capacity fluctuation of virtual machine. Therefore, the expected computing capacity of virtual machine is employed to calculate task computing time, which fully takes the influence of computing capacity fluctuation of virtual machine into account. Besides, according to task type, data-intensive tasks are allocated to the same virtual machine as much as possible, so as to reduce the data transmission time.

The directed acyclic graph model [7] is used to represent workflow. Before describing the static scheduling algorithm, some definitions are firstly given.

Expected Computing Time (ECT): It is the average time when task t_i with type T_k is computed on virtual machine with type VT_k, which is shown as (5), where $ts(t_i)$ is the computing amount of task t_i, $c(v_j)$ is the expected computing capacity of virtual machine v_j according to (4), and VS_k is the set of virtual machines belonging to type VT_k.

$$ECT(t_i) = \sum_{j=1}^{|VS_k|} (ts(t_i)/c_e(v_j)) \Big/ |VS_k| \tag{5}$$

Earliest Start Time (EST) and Earliest Finish Time (EFT): When task t_i receives the data from all the predecessors, the task starts to execute, so EST is shown as (6), where $TT_{v(t_p),v(t_i)}$ is the data transmission time from virtual machine where task t_p is deployed to virtual machine where task t_i is deployed, and $pre(t_i)$ is the set of predecessors of task t_i. EFT is the finish time of task t_i that starts at its EST, shown as (7), where $suc(t_i)$ is the set of successors of task t_i.

$$EST(t_i) = \max_{t_p \in pre(t_i)} (EST(t_p) + ECT(t_p) + TT_{v(t_p),v(t_i)}) \tag{6}$$

$$EFT(t_i) = EST(t_i) + ECT(t_i) + \max_{t_s \in suc(t_i)} (TT_{v(t_i),v(t_s)}) \tag{7}$$

Latest Start Time (LST): It is derived from the latest start time of the end task in workflow, shown as (8), where $wf_{dl} - ECT(t_e)$ is LST of the end task t_e, and wf_{dl} is the deadline of workflow.

$$LST(t_i) = \max_{t_s \in suc(t_i)} (LST(t_s) - ECT(t_i) - TT_{v(t_i),v(t_s)}) \tag{8}$$

The difference between $LST(t_i)$ and $EST(t_i)$ is the Elastic Delay Time Remaining (EDTR). From the starting task to the end task in workflow, the paths with the longest finish time becomes critical paths. The tasks on the paths are called critical tasks, that is, EDTR of these tasks is smaller, and other tasks are non-critical tasks.

Based on these definitions, the static scheduling algorithm is described as algorithm 1.

Algorithm 1: the static scheduling

Input: workflow W, virtual machine set
Output: initial virtual machine allocation for tasks

1: initialize $VS_k \leftarrow \phi$, $VS'_k \leftarrow \phi$, $VS(pre_d(t_i)) \leftarrow \phi$;
2: for each task t_i in workflow W do
3: determine task type T_k of task t_i according to the duration of computing time or data transmission time;
4: find all virtual machines of type VT_k to form the set $VS_k = \{v_1, v_2, \cdots, v_{n_k}\}$;
5: for each virtual machine v_j in VS_k do
6: if the EFT of task t_i running on virtual machine v_j meets $EFT(t_i, v_j) \le EFT(t_i)$ then
7: $VS'_k \leftarrow VS'_k \bigcup v_j$;
8: end if
9: end for

10: if type T_k of task t_i is CT task then
11: case 1: if the predecessors are all CT tasks, then the virtual machine is randomly selected from VS'_k for task t_i;
12: case 2: if there are single, multiple or all DT tasks in the predecessors, record the DT tasks as $pre_d(t_i)$, then for $v_s \in VS'_k$, the virtual machine v_{s^*} meeting $v_{s^*} = \arg\min_{v_s} \sum_{t_j \in pre_d(t_i)} TT_{v(t_j),v_s}$ is selected for task t_i;
13: case 3: if the predecessors are all BT tasks or include BT and CT tasks, record BT tasks as $pre_b(t_i)$, then for $v_s \in VS'_k$, the virtual machine v_{s^*} meeting $v_{s^*} = \arg\min_{v_s} \sum_{t_j \in pre_b(t_i)} TT_{v(t_j),v_s}$ is selected for task t_i;
14: case 4: if the virtual machine is not found from VS'_k, then the virtual machine closest to $EFT(t_i)$ is selected from VS'_k for task t_i;
15: end if
16: if type T_k of task t_i is DT task then
17: case 1: if the predecessors are all CT tasks, then a virtual machine is randomly selected from VS'_k for task t_i;
18: case 2: if there are single, multiple or all DT tasks in the predecessors, then for $v_s \in VS'_k \bigcap VS(pre_d(t_i))$, the virtual machine v_{s^*} meeting $v_{s^*} = \arg\max_{v_s} \sum_{t_j \in pre_d(t_i)} TT_{v(t_j),v_s}$ is selected for task t_i;.
19: case 3: if the predecessors are all BT tasks or include BT and CT tasks, then for $v_s \in VS'_k \bigcap VS(pre_b(t_i))$, the virtual machine v_{s^*} meeting $v_{s^*} = \arg\min_{v_s} \sum_{t_j \in pre_b(t_i)} TT_{v(t_j),v_s}$ is selected for task t_i;
20: case 4: if the virtual machine is not found from VS'_k, then the virtual machine closest to $EFT(t_i)$ is selected from VS'_k for task t_i;
21: end if
22: end for

B. The Dynamic Scheduling

In the execution of workflow, due to the computing capacity fluctuation of virtual machine and various failures, task finish time may change, resulting in the change of task nature. So, these changes need to be handled and different fault-tolerant strategies should be selected adaptively to respond to the change.

On the one hand, because virtual machine has the computing capacity fluctuation, it is necessary to estimate task finish time before task execution and judge whether task nature has changed. Assume that task t_i is assigned to virtual machine v_j, and task t_k is the task assigned to v_j and completed just before t_i, then the latest computing capacity $c_n(v_j)$ of v_j is $ts(t_k)/ACT(t_k)$, where $ACT(t_k)$ is the actual finish time of t_k. In addition, in order to alleviate the impact of jumping points, previous estimated value $c_o(v_j)$ and expected computing capacity $c_e(v_j)$ are added. Therefore, the estimated computing capacity $c_r(v_j)$ of v_j is shown as (9), where α and β are the weight values.

$$c_r(v_j) = \alpha \cdot c_n(v_j) + \beta \cdot c_o(v_j) + (1-\alpha-\beta) \cdot c_e(v_j) \tag{9}$$

On the other hand, during the execution of workflow, when task fails, task finish time may be delayed, resulting in the task nature of the subsequent tasks change. Therefore, in the dynamic scheduling, it should not only to deal with the faults in time, but also to adjust the fault-tolerant strategies for the subsequent tasks. According to task nature and fault type, the adaptive fault-tolerant strategies for tasks are selected. Because critical tasks can seriously affect the entire finish time of workflow, they should not be delayed. Therefore, the Primary-Backup strategy is used for critical tasks to ensure task finish. To the fault-tolerant strategies for non-critical tasks, the following six cases are mainly discussed as follows: 1) when the virtual machine v_j or host where task t_i is deployed fails temporarily during computing or data transmission, the task will continue to execute after the host or virtual machine is restored. Then $AST(t_i,v_j) + ACT(t_i,v_j) + RT_v + \max_{t_s \in suc(t_i)} TT_{v(t_i),v(t_s)}$ is the actual finish time $AFT(t_i,v_j)$ of t_i, where $AST(t_i,v_j)$ is the actual start time of task, $ACT(t_i,v_j)$ is the actual computing time of task, and RT_v is the recovery time of host or virtual machine. 2) when the communication of virtual machine v_j or host where task t_i is deployed fails temporarily during data transmission, the task will continue to execute after the network communication is restored. Then the $AFT(t_i,v_j)$ of t_i is $AST(t_i,v_j) + ACT(t_i,v_j) + \max_{t_s \in suc(t_i)} (TT_{v(t_i),v(t_s)} + RT_c)$, where RT_c is the recovery time of network communication. 3) when the virtual machine v_j where the task t_i is deployed fails permanently during data transmission, the task is resubmitted on the virtual machine v_n of another host. Then the $AFT(t_i,v_n)$ of task t_i on the new virtual machine v_n is $FST(t_i,v_j) + \max_{t_p \in pre(t_i)} TT_{v(t_p),v_n} + ACT(t_i,v_n) + \max_{t_s \in suc(t_i)} TT_{v_n,v(t_s)}$, where $FST(t_i,v_j)$ is the time when the fault occurred on virtual machine v_j. 4) when the communication between the virtual machine v_j where task t_i is deployed and the virtual machine where the successor task $t_{s'}$ is deployed fails permanently during data transmission, the data transmission is carried out through the connection bridge virtual machine v_b, and then $AST(t_i,v_j) + ACT(t_i,v_j) + \max(\max_{t_s \in suc(t_i)-t_{s'}} TT_{v(t_i),v(t_s)}, TT_{v(t_i),v_b} + TT_{v_b,v(t_{s'})})$ is the $AFT(t_i,v_j)$ of task t_i. 5) when the communication fails during data transmission, which is caused by the permanent fault of the virtual machine where the successor task $t_{s'}$ is deployed, it can be divided into two cases. If task $t_{s'}$ is non-critical task, the data is sent to the virtual machine where the

resubmitted task is deployed, otherwise, the data is sent to the virtual machine where the backup of task $t_{s'}$ is deployed. The $AFT(t_i, v_j)$ of task t_i is the finish time of data transmission to the virtual machine where the resubmitted task or the backup task is deployed. 6) when the virtual machine v_j where task t_i is deployed fails permanently during computing, the task is resubmitted to the virtual machine v_n of another host, and the $AFT(t_i, v_j)$ of task is the same as case 3).

In summary, the dynamic scheduling algorithm is described in algorithm 2.

Algorithm 2: the dynamic scheduling

Input: the result of the static scheduling
Output: the finish of workflow

```
for each task t_i in workflow W do
    estimate the computing time of task t_i according to (9);
    determine the nature of task t_i according to the actual finish time of the predecessors and the estimated computing time of task t_i;
    if task t_i is critical task then
        adopt the PB fault-tolerant strategy for task t_i;
    else set the fault-tolerant strategies of non-critical task for task t_i according to the six cases;
    end if
    if task t_i occurs fault
        handle the fault according to the six cases;
        compute the actual finish time of task t_i;
    end if
end for
calculate the finish time of the workflow and record the execution information of the workflow.
```

VII. EXPERIMENTS

A series of experiments are conducted by Java to evaluate the performance of the WSAFSS method and the impact factors on the performance are analyzed in this section. In experiments, the computing capacity of virtual machine is randomly selected from 250MIPS, 500MIPS, 750MIPS, 1000MIPS to 1250MIPS. The bandwidth is randomly selected from 500Mbps, 750 Mbps, 1000Mbps, 1250Mbps to 1500 Mbps. The computing capacity fluctuation of virtual machine is established according to the proposed fluctuation model. The DAG generator used in [16] is selected to create workflows, and the deadline of workflow is set as 1.3 times of the finish time of workflow without failures. The WSAFSS method is compared with the WSAFSS without task classification (NTCWSAFSS), the WSAFSS without fault-tolerant strategy adjustment (NFSAWSAFSS) and DFTWS [11]. In addition, completion ratio of workflow (WCR) and completion time of workflow (WCT) are used to measure the performance of these methods, where WCR is the percentage of workflows which successfully complete before their deadlines, and WCT is the average time consumed to complete one workflow.

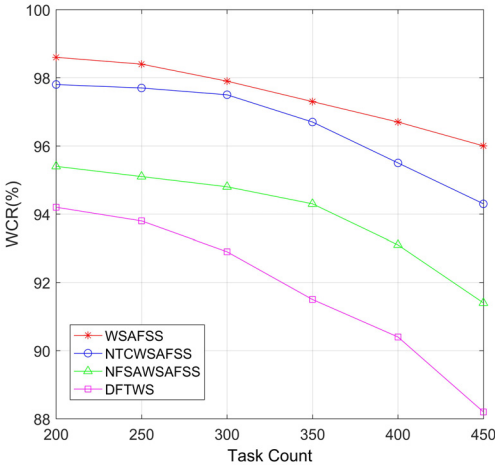


Figure 2. The WCR with the increasing task count

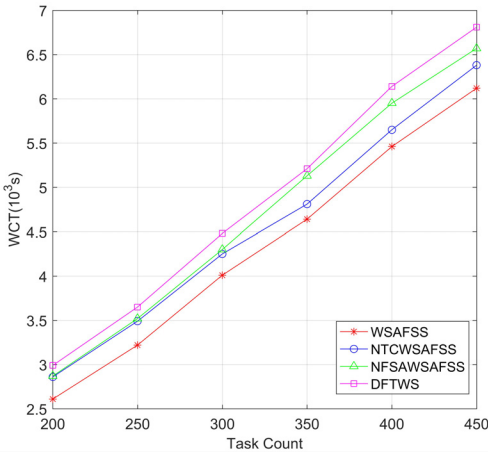


Figure 3. The WCT with the increasing task count

A. *The Impact of the Task Count*

When the fault ratio is 0.06 and the computing capacity fluctuation ratio is 0.12, the impact of the task count is shown in Fig. 2 and Fig. 3. Fig. 2 reveals that when the task count increases, WCR in all methods decreases, but WCR in WSAFSS is kept at about 0.97 and higher than other methods. Meanwhile, Fig.3 displays that WCT in all methods increases, and WCT in WSAFSS is lower than other methods. It is remarkable that the more tasks, the more obvious the effect. This is because that WSAFSS adopts dynamic fault-tolerant strategy and has the perception of fluctuation, which avoids the delay of critical tasks to a certain extent, especially for the workflow with more tasks. Besides, the more tasks in the workflow, the greater the probability of task failure, which leads to the rapid performance degradation in NFSAWSAFSS and DFTWS. In particular, DFTWS cannot perceive fluctuation, resulting in the longest WCT and the lowest WCR.

B. *The Impact of the Fault Ratio*

When the task count is 450 and the computing capacity fluctuation ratio is 0.12, the impact of the fault ratio is shown in Fig.4 and Fig.5. Fig.4 manifests that when the fault ratio

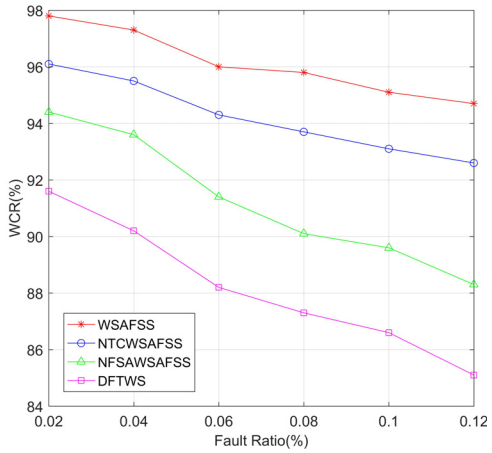


Figure 4. The WCR with the increasing fault ratio

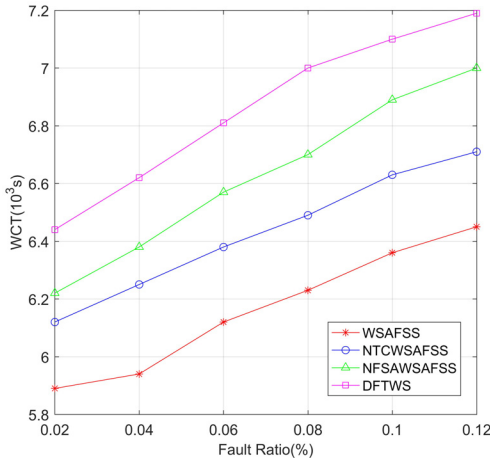


Figure 5. The WCT with the increasing fault ratio

increases, WCR in all methods decreases, but WCR in WSAFSS is significantly higher than other methods. Meanwhile, Fig.5 shows that WCT in all methods increases, but WCT in WSAFSS is significantly lower than other methods. This is because that, when task failure increases, a large number of tasks are delayed, which changes the nature of the subsequent tasks in the workflow. WSAFSS can capture the changes well, and adopt the PB strategy for critical tasks, so as to alleviate the delay of tasks to a certain extent and improve WCR and WCT. However, NFSAWSAFSS and DFTWS cannot adjust the fault-tolerant strategies for tasks in real time. Besides, DFTWS cannot capture the fluctuation, so the efficiency of the two methods is relatively lower.

C. *The Impact of the Fluctuation*

When the task count is 450 and the fault ratio is 0.06, Fig.6 and Fig.7 show the impact of the computing capacity fluctuation. Fig.6 displays that when the fluctuation ratio increases, WCR in all methods decreases, but WCR in WSAFSS is higher than other methods. Fig.7 reveals that WCT in all methods increases, but WCT in WSAFSS is lower than other methods. This is because that WSAFSS and NTCWSAFSS consider the delay of task computing time caused by the computing capacity fluctuation of virtual machine in task scheduling. Meanwhile, the two methods can change the fault-tolerant strategy for tasks in real time, so WCR and WCT are better than other methods. However, DFTWS cannot perceive the fluctuation, and cannot adjust the fault-tolerant strategies for tasks in real time, which leads to the lowest efficiency of this method. Although NFSAWSAFSS takes the fluctuation into account, but it lacks the measures to adjust the fault-tolerant strategies for tasks, which leads to relatively lower efficiency than WSAFSS and NTCWSAFSS.

D. *The Impact of the Fault Ratio and the Fluctuation*

The impact of the fault ratio and the computing capacity fluctuation is shown in Fig.8 and Fig.9, when the task count is 450. It is shown that WSAFSS maintains high workflow completion rate and short workflow completion time under different fluctuation ratio and fault ratio, which fully proves the stability and efficiency of this method. This is because the method not only considers the scheduling of tasks of various types, but also pays attention to the uncertainty of cloud environment. And it makes a reasonable response to the change of task nature in the process of workflow execution.

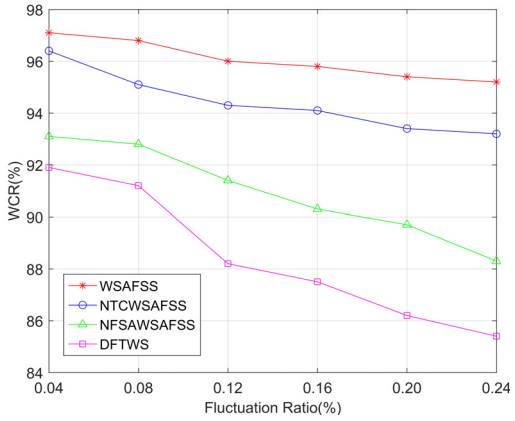


Figure 6. The WCR with the increasing fluctuation ratio

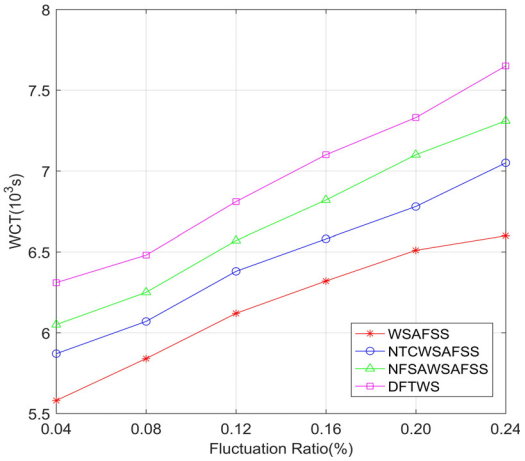


Figure 7. The WCT with the increasing fluctuation ratio

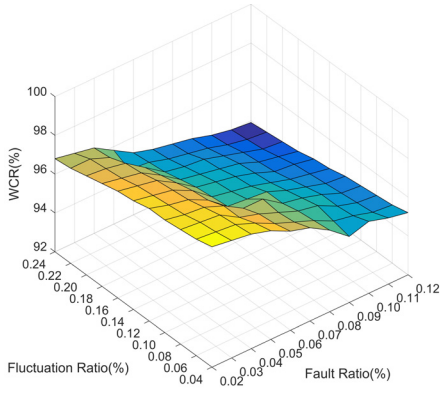


Figure 8. The WCR with fault ratio and fluctuation ratio

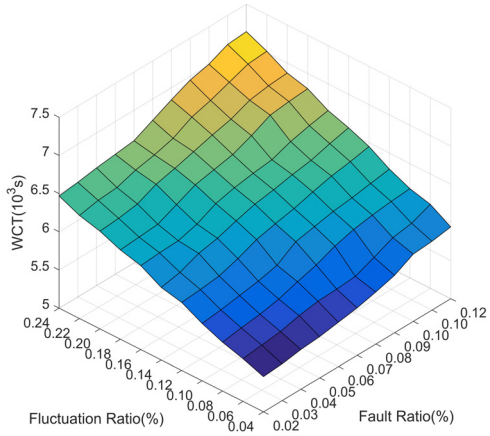


Figure 9. The WCT with fault ratio and fluctuation ratio

VIII. CONCLUSION

In this paper, a dynamic WSAFSS method for workflow scheduling is proposed. In the method, tasks and virtual machines are divided into six categories. Based on the classification, the workflow scheduling not only considers the tasks of various types, but also makes full use of the advantages of various virtual machines. Besides, the proposed method focuses on the uncertainty in cloud environment, mainly including the execution faults of tasks and the computing capacity fluctuation of virtual machine. Concretely, an extended fault-tolerant model based on virtual machine fault and data transmission fault is proposed, which effectively solves all kinds of faults during task execution. Meanwhile, in order to describe the impact of the computing capacity fluctuation of virtual machine during task execution, a novel computing capacity fluctuation model of virtual machine based on multiple states and jump points is proposed. Finally, in the uncertain cloud environment, a dynamic workflow scheduling algorithm is proposed, which seamlessly combines multiple tasks, multiple virtual machines, fault-tolerant model and fluctuation model. Experimental results reveal that the proposed method is capable to achieve higher WCR and shorter WCT than other methods under different impact factors.

ACKNOWLEDGMENT

This research is supported by the Natural Science Project of Henan Science and Technology Department (Grant no. 182102210388).

REFERENCES

[1] X. Zhu, J. Wang, H. Guo, D. Zhu, L. T. Yang, and L. Liu, "Fault-tolerant scheduling for real-time scientific workflows with elastic resource provisioning in virtualized clouds," IEEE Trans. Parallel. Distrib. Syst., vol. 27, no. 12, pp. 3501-3517, December 2016

[2] H. Raei, and N. Yazdani, "Performability analysis of cloudlet in mobile cloud computing," Inf. Sci., vol.388, pp.99-117, May 2017

[3] R. Buyya , C. S. Yeo , S. Venugopal, J. Broberg, I. Brandicet, "Cloud computing and emerging IT plat- forms: Vision, hype, and reality for delivering computing as the 5th utility," Future Gener. Comput. Syst., vol.25, no.6, pp.599-616, June 2019

[4] H. Chen, F. Z. Wang, and N. Helian, "Entropy4Cloud: Using Entropy-Based Complexity to Optimize Cloud Service Resource Management," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp.13-24, February 2018

[5] M. Masdari, S. Valikardan, Z. Shahi, and S. I. Azar, "Towards workflow scheduling in cloud computing: A comprehensive analysis," Journal of Network and Computer Applications, vol. 66, no. C, pp. 64-82, May 2016

[6] G. Yao, Y. Ding, L. Ren, K. Hao, and L. Chen, "An immune system-inspired rescheduling algorithm for workflow in Cloud systems," Knowledge-Based Systems, vol.99, no. C, pp. 39-50, May 2016

[7] Y. Ding, G. Yao, and K. Hao, "Fault-tolerant elastic scheduling algorithm for workflow in Cloud systems," Inf. Sci., vol. 393, no. 2017, pp. 47-65, July 2017

[8] H. Han, W. Bao, X. Zhu, X. Feng, and W. Zhou, "Fault-Tolerant Scheduling for Hybrid Real-Time Tasks Based on CPB Model in Cloud," IEEE Access, vol. 6, no. 2018, pp. 18616-18629, February 2018

[9] K. Vinay, S. M. Dilip Kumar, S. Raghavendra, and K. R. Venugopal, "Cost and fault-tolerant aware resource management for scientific workflow using hybrid instances on clouds," Multimedia Tools and Applications, vol. 77, no. 8, pp. 10171-10193, October 2018

[10] J. Lee, and J. Gil, "Adaptive fault-tolerant scheduling strategies for mobile cloud computing," Journal of supercomputing, vol. 75, no. 8, pp. 4472-4488, January 2019

[11] N. Wu, D. Zuo, and Z. Zhang, "Dynamic Fault-Tolerant Workflow Scheduling with Hybrid Spatial-Temporal Re-Execution in Clouds," Information, vol. 10, No. 2019, pp.e169, May 2019

[12] L. Zhang, L. Zhou, and A. Salah, "Efficient scientific workflow scheduling for deadline-constrained parallel tasks in cloud computing environments," Inf. Sci., vol.531, pp. 31-46, August 2020

[13] H. Chen, X. Zhu, H. Guo, J. Zhu, X. Qin, and J. Wu, "Towards to energy-efficient scheduling for real-time tasks under uncertain cloud computing environment," The Journal of Systems and Software, vol. 99, no. 2015, pp. 20-35, January 2015

[14] H. Yan, X. Zhu, H. Chen, H. Guo, W. Zhou, and W. Bao, "DEFT: Dynamic Fault-Tolerant Elastic scheduling for tasks with uncertain runtime cloud," Inf. Sci., vol. 477, no. 2019, pp. 30-46, March 2019

[15] N. Malakoutifar, and H. Motallebi, "Task graph scheduling in the presence of performance fluctuations of computational resources," Turkish Journal of Electrical Engineering and Computer Sciences, vol. 27, no. 3, pp. 2170-2185, May 2019

[16] Z Zhu, G Zhang, M Li, and X Liu. "Evolutionary Multi-Objective Workflow Scheduling in Cloud," IEEE Trans. Parallel. Distrib. Syst., vol. 27, no. 5, pp.1344-1357, June 2015

IP Geolocation Method based on Neighbor IP Sequences

Yong Gan
Zhengzhou Institute of Engineering and Technology
Zhengzhou, China
e-mail: ganyong@zzuli.edu.cn

Helin Zhang
School of Computer and Communication Engineering
Zhengzhou University of Light Industry
Zhengzhou, China
e-mail: zhanghelin5460@foxmail.com

Yuanbo Liu
School of Computer and Communication Engineering
Zhengzhou University of Light Industry
Zhengzhou, China
e-mail: zzulilyb@163.com

Lei He
School of Computer and Communication Engineering
Zhengzhou University of Light Industry
Zhengzhou, China
e-mail: helei@zzuli.edu.cn

***Abstract*—IP geolocation aims at determining the geographic location of an Internet host, which can improve the performance and security of the Internet application, and bring about novel services. Existing methods usually use measurement-based technology or data analysis-based technology, and less consider the relationship between IP addresses，this paper proposed an IP geolocation approach based on neighbor sequences. Considering the aggregation characteristics of IP addresses, the IP address location library and mobile traffic data are taken as the original data. The neighbor sequences of IP addresses are first calculated, then converted into corresponding latitude and longitude sequences, and then the model is built and solved. The experimental results based on IP addresses in Henan Province, China show that the geographic location of IP addresses can be determined by neighbor IP sequences with the mean error is between 20 kilometers and 30 kilometers. This method can also be combined with other methods to obtain better results.**

Keywords-component; IP geolocation; neighbor sequences; latitude and longitude

I. INTRODUCTION

At user device does not provide a GPS location technology, using the IP address geolocation techniques is the preferred method for determining the location of a user. Currently, for the increasing size of the Internet user base, a large number of Internet services are required to obtain location information of the user. How to accurately determine a user's geographic location has become an Internet application in a significant issue.

Existing IP address geolocation technologies can be divided into two categories: 1) Based on data analysis methods, such methods use non-measurement technology, through the processing and analysis of WHOIS database, web pages and other related data, to obtain the geographic location corresponding to the IP address. 2) Measurement-based methods, whose principle is to obtain network topology, route hops, or delay detection of targets through traceroute, to infer the geographic location of the IP address. Such methods can be further subdivided into methods based on space theory and methods based on probability estimation [1], mainly including Geoping [2], TBG [3], Poist [4] and so on.

In IP address geolocation technology based on data analysis, the data can be divided into structured data, semi-structured data, and unstructured data according to the degree of structure of the data used. Registration and filing records are common structured data containing IP addresses (segments) and their corresponding geographic information [5], such as WHOIS database, DNSLOC records, DNS names, etc. These data can be used to infer the geographic location of IP addresses.

Wang et al. [1] divided geolocation methods based on registration records into three categories: 1) inferring network structure and database information. It may appear extensive deviation method based on geographic positioning registration records, because some large entities may have multiple servers scattered in different places, but the domain name is registered to the same address [4]; 2) querying the WHOIS database; 3) measuring hostnames and inferring based on database information. In semi-structured data, geographic information may exist as an attribute of an object in the data. Dan et al. [5] extracted IP addresses and GPS coordinates from mobile device search engine logs to construct the most extensive set of ground truth values (Ground Truth) to date, with 8.4 million IP address geolocation records. Web pages are the most common unstructured data. You can use text mining [6], data mining, and statistics [7] to extract geographic information from massive Web pages and associate them with specific IP addresses (segments) to achieve IP geolocation. Guo et al. [8] proposed the Structon method, which combines Web mining, inference and IP traceroute to reach a more accurate IP address location. Backstrom et al [9] proposed a method based on a social map, using the position of the user's friends to determine the user's location, and achieved a good positioning effect.

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00016

Guo et al. [8] found that network administrators tend to assign consecutive IP address segments to the same area or similar locations, which means that consecutive IP addresses tend to cluster together in geographical distribution, that is, consecutive IP addresses tend to be neighbor geographically. This paper refers to this assumption as the aggregation characteristics of IP addresses. Based on this assumption, if the geographical location information of other IP addresses neighbor to an unknown location IP address can be obtained, based on the geographic location of these neighbor IP addresses, the physical spatial location range of the target IP address can be inferred.

Based on the above assumptions, this paper proposes an IP address geolocation method based on neighbor sequences. First calculate the neighbor sequences of the IP address and convert it to the corresponding latitude and longitude, then build the model and solve it. This paper uses the following two types of data to verify the method: 1) The open-source IP positioning database [10] contains the IP address segment and its corresponding administrative division address; 2) The experimental data of a province in China sampled in the education and scientific research network, this data is the network traffic of the mobile application. The traffic contains the IP address, latitude and longitude, and forms a corresponding relationship. The experimental results show that the geographical location of the IP address can be determined by the neighbor IP sequences, and the average positioning error is 20-30 kilometers, which realizes the positioning at the district and county level. This method provides a new solution to the problem of IP address geolocation. Simultaneously, it can also be combined with other methods based on measurement or data analysis to obtain better results. The basic principle of the IP address geolocation method based on neighbor sequences proposed in this paper is shown in Figure 1.

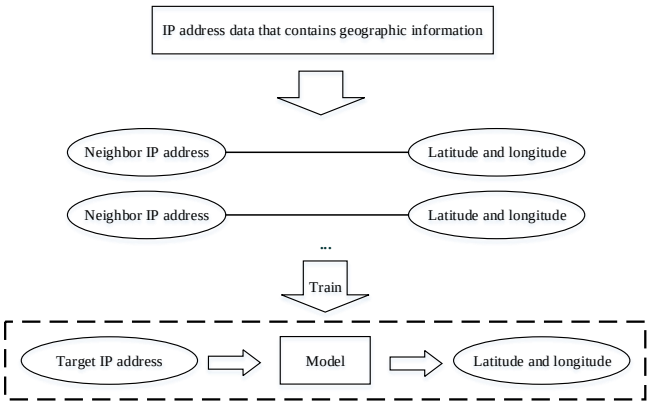


Figure 1. The basic principle of geolocation

II. RELATED DEFINITIONS

The common representation of IP address [11, 12] is 4 decimal numbers separated by English ".", which is essentially a 32-bit binary integer, which can be expressed as $IP = b_{32}b_{31}\cdots b_1$, where $b_i(1 \leqslant i \leqslant 32)$ It is a binary number 0 or 1, and mathematical operations can be performed.

In order to measure the proximity of IP addresses, this paper needs to define the distance between two IP addresses. The definition of IP distance should meet the following two conditions:

1) The distance between two identical IP addresses is 0; 2) Between different IP addresses, the longer the same prefix, the closer the distance. This property ensures that the IP address distance in the same network is always smaller than the IP address distance between different networks.

Definition 1 (IP distance) IP distance is defined as:

$$d(p_1, p_2) = \begin{cases} 0 & , p_1 = p_2 \\ length(p_1 \oplus p_2), otherwise \end{cases} \quad (1)$$

Among them, Length (x) represents the number of bits of a binary number x without leading 0. In the result of the XOR operation, the same bit in the two IP addresses becomes 0, and the different bit becomes 1, that is, the same prefix of the two IP addresses is a string of 0 in the result, and is located in the high order of the number. Therefore, the Length function can measure the length of the longest common prefix in the two IP addresses, and the longer the longest common prefix, the smaller the function value. Therefore, Definition 1 satisfies the two conditions required for the definition of IP distance.

For example, for the three IP addresses listed in Table 1, d(A,B)=4 and d(A,C)=20. This indicates that the distance between AB is smaller than that of AC, which is consistent with the IP address decimal.

TABLE I. EXAMPLES OF IP ADDRESS AND CORRESPONDING BINARY FORMAL

Number	IP address	Binary formal
A	115.158.71.170	01110011100111100100011110101010
B	115.158.71.161	01110011100111100100011110100001
C	115.84.129.64	01110011100000010111111101000000

Definition 2 (Proximity IP Set and Proximity IP Address) For the specified IP address p , any IP address set with a distance n from p is denoted as $S_{p,n}$, which is called the n-proximity IP set of p. The elements of $S_{p,n}$ are called the n-proximity IP address of p, and are denoted as $N_{p,n}$. For a given IP address p, its neighbor IP set $S_{p,n}$ is determined. According to the definition of IP distance, the binary number of XOR of any element in $S_{p,n}$ and p is n. This means that the high-order bits of these elements are the same as p, the n-th b_n is different from p, and the value of the lowest $n-1$ bit $b_i(1 \leqslant i \leqslant n-1)$ can be any 0 or 1. Therefore, $S_{p,n}$ can be represented by the closed range $[N1_{p,n}, N2_{p,n}]$ of the IP address, where $N1_{p,n} = b_{32}\cdots b_{n0}\cdots 0$, $N2_{p,n} = b_{32}\cdots b_{n1}\cdots 1$.

Definition 3 (neighbor sequences) Given the minimum distance n_s and maximum distance n_e of the IP address p, the sequences of IP addresses $N_{p,n_s}, N_{p,n_{s+1}}, \cdots, N_{p,n_e}$ is called the neighbor sequences of p.

III. DESCRIPTION OF GEOLOCATION ALGORITHM

The goal of IP address positioning is that given an IP address, the location information of its physical space can be quickly queried, including precise longitude and latitude and coarse-grained administrative regionalized addresses [13].

Based on the aggregation characteristics of IP addresses, this paper proposes a geolocation method for IP addresses based on neighbor sequences. This method infers the location range of the target IP address based on the latitude and longitude of multiple neighbor IPs. To this end, first need to have a mapping relationship table from IP address to latitude and longitude as the benchmark input of the algorithm.

We have two different kinds of original data: 1) Open source IP address location database, each record is in the form of six tuples (starting IP address, ending IP address, country, province, city, district), of which districts and counties It may be empty; 2) The mobile application network traffic of a province in China extracted from the China Education and Research Network can be parsed from the data to obtain triples of the form (IP address, longitude, latitude). Using Baidu's geocoding API [14], the administrative geographic address of the IP address location database is encoded into latitude and longitude to form a quadruple (starting IP address, ending IP address, longitude, latitude).

For mobile application network traffic, the IP addresses of each record are aggregated, and finally each IP address gets a set of latitude and longitude, and the maximum distance between these latitude and longitude is calculated. If the maximum distance is more than 20 kilometers, this IP address is considered to be dynamically allocated and the allocated addresses are scattered, which is not considered in this paper. After discarding the data with the maximum distance above 20 kilometers, take the collection center of these latitude and longitude as the final latitude and longitude, and record the number of aggregation. Figure 2 shows the flow chart of the algorithm.

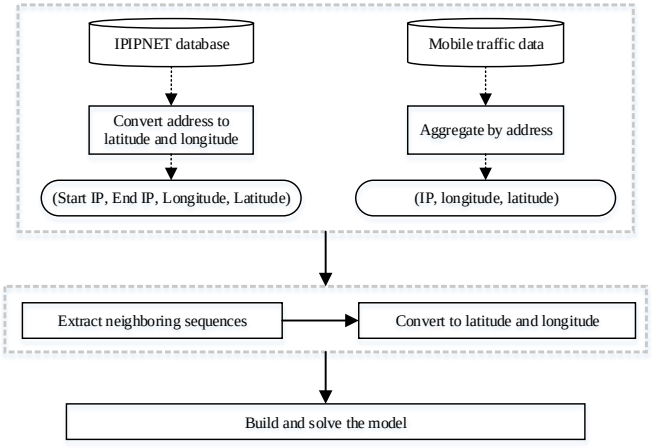


Figure 2. Flowchart of algorithms

A. Construct the Latitude and Longitude Corresponding to the Neighbor Sequences

According to definition 2 and formula (1), for the target IP address p and the IP distance n, the calculation formula of the neighbor set $S_{p,n}$ is:

$$S_{p,n} = \begin{cases} [p, p] & , n = 0 \\ [N1_{p,n}, N2_{p,n}], & otherwise \end{cases} \quad (2)$$

For each IP address p in S_a, calculate n_e-n_s+1 neighbor IP sets $[N1_{p,n}, N2_{p,n}]$ and then traverse S_a and select in the neighbor set IP address $N_{p,n}$, if there is no IP address that satisfies the condition, then traverse the set S_b to find the longest IP address segment that intersects with the neighbor set. Finally, map these neighbor IP addresses (segments) to the corresponding latitude and longitude. Table 2 lists the related parameters of algorithm and their description. The algorithm description is as follows:

TABLE II. DESCRIPTION OF PARAMETERS

Parameter	Explanation
n_s, n_e	The minimum distance and maximum distance between the neighbor IP and the target IP
$N_{p,n}$	IP address with distance n from IP address p
S_a, S_b	Respectively represent the processed mobile application network traffic data collection and IPIPNET data collection
S_{train}, S_{test}	IP address training set and test set
T	The set of latitude and longitude sequences corresponding to neighbor sequences

Calculate the latitude and longitude of neighbor sequences:

Input: S_a, S_b, n_s, n_e
Output : Set of latitude and longitude series T corresponding to neighbor sequences

BEGIN
T= Ø
FOR $p \in S_a$ DO
R= Ø
FOR $n \leftarrow n_s$ TO n_e DO
Calculate the neighbor IP set $[N1_{p,n}, N2_{p,n}]$ with distance p of n, select IP set M in S_a in the range of $[N1_{p,n}, N2_{p,n}]$
IF Size(M)=1 THEN
R $[n]$ ← Latitude and longitude of the element of M
ELSE IF Size(M)≠0 THEN
R $[n]$ ← The latitude and longitude of the element with the largest aggregation number of M
ELSE
Select the IP segment L with the longest intersection with $[N1_{p,n}, N2_{p,n}]$ in S_b

```
            R [n] ← Latitude and longitude of the element of L
          END
        END
            T[p] ← R
    END
    RETURN T
END
```

B. Deduce the Latitude and Longitude of the Target Address

Based on the above process, the IP address location problem can be transformed into using the latitude and longitude corresponding to the neighbor sequences to infer the latitude and longitude of the target IP address. Taking the longitude and latitude sequences X_i as input, and the longitude and latitude Y^i of the target IP address as output :

$$Y^i = f(X_i) \tag{3}$$

The actual task is to guess the function *f*. If *Y* and *X* are linear, then:

$$Y^i = \omega_T X_i \tag{4}$$

Where Y^i is a 2-dimensional row vector, ω is an n_e-n_s+*1*-dimensional column vector, and X_i is a matrix of (n_e-n_s+*1*)*2. In order to learn this function, the loss function is defined as:

$$Loss(X) = \sum_{i=1}^{m} (Y^i - Y_i) \bullet (Y^i - Y_i) \tag{5}$$

Where Y_i is the true value of the i-th set of data X_i. And the requested model is as follows

$$\omega = \arg\min_{\omega} Loss(X) \tag{6}$$

This paper chooses the method of mini-batch gradient descent [15] to solve the model. Compared with batch gradient descent and stochastic gradient descent, the speed is faster than the batch gradient descent, and the accuracy is better than the random gradient descent.

The main parameters of the algorithm are the minimum distance n_s and the maximum distance n_e of neighbor sequences. When the IP distance reaches a large value, there is no significant correlation between the geographical locations of the two. At the same time, when the IP distance is small enough, it can be assumed that both are in the same small LAN. However, when determining the geographic location of the target IP address, you cannot only consider neighbor sequences with a sufficiently small distance. IP address pairs with very small distances from each other will not appear in the data in large numbers. Relying on such occasional data will lead to overfitting of the model.

IV. Experimental Results and Analysis

Based on the existing data, this paper selects different n_s and n_e for comparison. Considering that shorter neighbor sequences are more susceptible to noise data, the length of the neighbor sequences selected in this paper is 8-16. Table 3 lists the accuracy of the algorithm under different parameters.

TABLE III. Performance comparison of algorithms with different parameters of parameters

n_s	n_e	average error / kilometer	median / kilometer	variance	Minimum error / kilometer	Maximum error / kilometer
8	16	26.86	16.53	827.17	0.35	210.58
8	20	26.47	14.45	931.25	0.37	245.30
8	24	27.97	17.15	1195.51	0.40	266.90
9	16	27.87	18.76	1048.57	0.07	245.22
9	20	26.51	17.99	1045.82	0.11	238.36
9	24	28.44	18.74	1045.85	0.49	254.11
10	17	28.75	14.39	1182.65	0.22	257.48
10	20	27.45	17.35	877.75	0.55	223.62
10	24	30.17	17.46	1111.18	0.15	232.11
11	18	28.31	19.12	955.34	0.07	220.86
11	20	31.06	23.08	969.17	0.18	226.94
11	24	29.35	21.06	952.20	0.10	223.86
12	19	32.94	24.04	1059.94	0.65	256.11
12	20	32.70	23.92	1079.66	0.55	225.44
12	24	32.20	21.72	1139.81	0.12	236.60

It can be seen from Figure. 3 and Figure. 4 that under different parameters, the distribution of errors shows the law that the error distance increases and the number of errors rapidly decreases. In addition, under different parameters, although the error distance increases and the number of errors decreases at different rates, the accuracy of the algorithm does not differ by orders of magnitude. This indicates that within the selected n_s and n_e ranges, the physical space position of the neighbor sequences has a greater correlation with the physical space position of the target IP address.

However, it can still be seen that when the parameters n_s =8 and n_e =20, compared to when n_s =8 and n_e =24, the number of errors under the same distance when the error distance is smaller, and the number of errors under the same distance when the error distance is larger. These two points make the average error and median error of the algorithm when n_s =8 and n_e =20 smaller, thus making the algorithm more accurate.

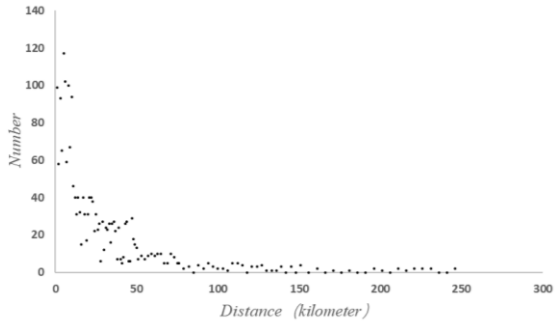


Figure 3. Error distribution of algorithm when n_s=8 and n_e=20

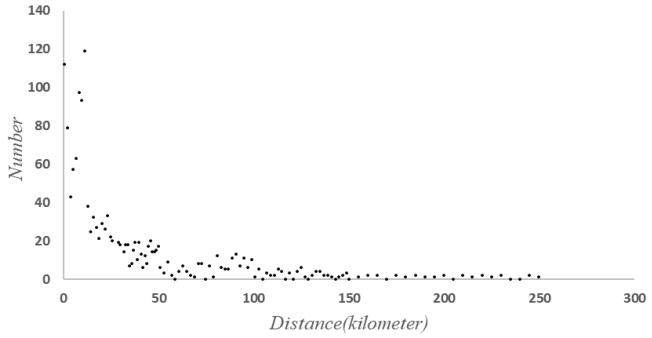


Figure 4. Error distribution of algorithm when n_s=8 and n_e=24

It can be observed from Table 3 and Figures 3 and 4 that n_s and n_e affect the accuracy of the algorithm. In actual use, due to different data, the specific distribution of data is also necessarily different, so experiments should be conducted on the basis of fully studying the distribution of data to determine the specific values of n_s and n_e to obtain the optimal positioning performance.

The average error of the algorithm is 20-30 kilometers, and the median error is about 20 kilometers, which basically realizes the positioning at the district and county level. Table 4 lists the positioning results of the algorithm and the results given by other IP address libraries. It can be seen that the results given by the algorithm have a finer granularity, and the accuracy is comparable to that of each database.

TABLE IV. GEOLOCATION RESULTS

IP address	Algorithm results	Baidu	ChunZhen
117.158.221.215	Zhongyuan District, Zhengzhou	Zhongyuan District, Zhengzhou	Zhengzhou
219.157.12.36	Luolong District, Luoyang	Hebi	Luoyang
218.196.192.25	Longting District, Kaifeng	Kaifeng	Kaifeng
42.239.250.215	Qibin District, Hebi	Hebi	Hebi
182.127.208.2	Xiping County, Zhumadian	Xiping County, Zhumadian	Zhengzhou
61.158.128.9	Luolong District, Luoyang	Kaifeng	Luoyang
175.106.255.25	Hualong District, Puyang	Hualong District, Puyang	Puyang
221.176.240.15	Qibin District, Hebi	Qibin District, Hebi	Hebi

V. CONCLUSION

Regarding the geographic location of IP addresses, existing data analysis-based methods pay more attention to the extension attributes of IP addresses, such as administrative division addresses and domain name registration records, and pay less attention to the relationship between IP addresses. This paper focuses on the relationship between IP addresses. According to the aggregation characteristics of IP addresses, a geolocation method for IP addresses based on neighbor sequences is proposed. First, select the neighbor sequences of the IP address and convert it to the corresponding latitude and longitude sequences, then use a linear model to fit the latitude and longitude sequences, and select the parameter that minimizes the loss function as the model. Experiments show that by selecting appropriate parameters, this method can give a more accurate physical location of the IP address. Limited by the data size of the experiment, the positioning error of this experiment is 20-30 kilometers on average. If the amount of data reaches a million level, this method is expected to limit the average positioning error to within 10 kilometers.

Starting from the relationship between IP addresses, this method considers the aggregation characteristics of IP addresses, provides a new idea for the IP address geolocation problem, and is a powerful complement to the existing IP address geolocation technology. Based on the data analysis method, if the relationship between IP addresses is introduced on the basis of the IP address extension attributes, a large amount of information can be added and more features can be obtained, then the model will have more parameters and the expression ability will also be enhanced, you can expect to obtain better results. Accurate IP address positioning results can effectively support related Internet industries, such as network security and advertising. This work is based on IPv4 addresses. Although IPv6 addresses are not widely used at present, they are growing rapidly. The 37th China Internet Development Report [16] shows that from 2014 to 2015, the annual growth rate of the number of IPv4 addresses in China reaching 9.6%, the next step will be dedicated to the geographic positioning of IPv6 addresses.

ACKNOWLEDGMENT

This work is supported by the Nation Nature Science Foundation of China (NSFC), (NO. 61572445), NSFC Joint Fund Key Project (NO. U1804263).

REFERENCES

[1] Jinxia, W. , Xiaoyan, X. , Min, Y. , & Tianning, Z. . (2016). IP Geolocation Technology Research Based on Network Measurement. Sixth International Conference on Instrumentation & Measurement. IEEE.

[2] Padmanabhan, V. N. , & Subramanian, L. . (2001). An investigation of geographic mapping techniques for internet hosts. Acm Sigcomm Computer Communication Review.

[3] Katz-Bassett, E. , John, J. P. , Krishnamurthy, A. , Wetherall, D. , & Chawathe, Y. . (2006). Towards IP geolocation using delay and topology measurements. Acm Sigcomm Conference on Internet Measurement. ACM.

[4] Eriksson, B. , Barford, P. , Maggs, B. , & Nowak, R. . (2012). Posit: a lightweight approach for ip geolocation. ACM SIGMETRICS Performance Evaluation Review, 40(2), 2-11.

[5] Ovidiu Dan, Vaibhav Parikh, & Brian D. Davison. (2016). Improving ip geolocation using query logs.

[6] Feitosa, R. M. , Labidi, S. , Santos, A. L. S. D. , & Santos, N. . (2013). Social Recommendation in Location-Based Social Network Using Text Mining. International Conference on Intelligent Systems. IEEE Computer Society.

[7] Yuan, T. , Cheng, J. , Zhang, X. , Liu, Q. , & Lu, H. . (2013). A Weighted One Class Collaborative Filtering with Content Topic Features. International Conference on Multimedia Modeling. Springer Berlin Heidelberg.

[8] Guo, C. , Liu, Y. , Shen, W. , Wang, H. J. , & Zhang, Y. . (2009). Mining the Web and the Internet for Accurate IP Address Geolocations. Infocom. IEEE.

[9] Backstrom, L. , Sun, E. , & Marlow, C. . (2010). Find me if you can: Improving geographical prediction with social and spatial proximity. Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010. ACM.

[10] Poese, I. , Uhlig, S. , Mohamed Ali Kâafar, Donnet, B. , & Gueye, B. . (2011). Ip geolocation databases: unreliable?. Acm Sigcomm Computer Communication Review, 41(2), 53-56.

[11] Tetsuya, S. , & Hiroyuki, O. . (2018). Ip address assignment device and ip address assignment method.

[12] Almohri, H. M. J. , Watson, L. T. , & Evans, D. . (2019). Predictability of ip address allocations for cloud computing platforms. IEEE Transactions on Information Forensics and Security, PP(99).

[13] Zu, S. , Luo, X. , Liu, S. , Liu, Y. , & Liu, F. . (2018). City-level ip geolocation algorithm based on pop network topology. IEEE Access, PP, 1-1.

[14] http://lbsyun.baidu.com/index.php?title=webapi/guide/webservice-geocoding.

[15] Khirirat, S. , Feyzmahdavian, H. R. , & Johansson, M. . (2017). Mini-batch gradient descent: Faster convergence under data sparsity. IEEE Conference on Decision & Control. IEEE.

[16] CNNIC. The 37th Statistical Report on Internet Development in China.http://www.cnnic.net.cn.

Constructing Formal Specification Models from Domain Specific Natural Language Requirements

Jun Hu
College of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics
Nanjing, China
hujun@nuaa.edu.com

Jiancheng Hu
College of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics
Nanjing, China
jiancheng.hu0930@qq.com

Wenxuan Wang
College of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics
Nanjing, China

Jiexiang Kang
Department of Software
China National Aeronautic Radio Electronics Research Institute
Shanghai, China

Hui Wang
Department of Software
China National Aeronautic Radio Electronics Research Institute
Shanghai, China

Zhongjie Gao
Department of Software
China National Aeronautic Radio Electronics Research Institute
Shanghai, China

***Abstract*—One important way to improve the quality of safety-critical software is to produce a good software requirement satisfying several key properties, such as: integrity, consistency, and well organized, etc. Our work is based on airborne software domain, and propose a framework to translate the software requirements, which are itemized with domain natural language in avionics, effectively into a formal specification model VRM (Variable Relation Model), which has table-style structures with formal semantics. Firstly, considering avionics domain characteristics, a domain concept library is established including different types of variables and concepts. Then, a set of domain-oriented requirements templates are defined, such as: general event/condition, display event/condition, etc. According to VRM model element semantics, three types model construction algorithms are designed to complete the translation automatically. And in the case study, the Engine Indication and Crew Warning System (EICAS) was selected to show how to construct formal models from natural language requirements.**

***Index Terms*—safety-critical software, VRM, EICAS, domain template library, model transition**

I. Introduction

Safety-critical software [1], [2] refers to a type of software used in safety-critical systems such as aviation, aerospace, transportation, and energy. These software systems require high safety, reliability, and robustness [3]. Errors or vulnerabilities caused by the software may lead to very serious consequences [4], [5], such as: heavy losses of property, severe damage to the environment or a large number of casualties, etc. In recent years, with the rapid growth of the number of functions and complexity of various safety-critical systems, how to effectively develop such software systems has become an great challenge in the area of safety-critical systems.

From the perspective of the software life cycle, it is very important to construct a product that meets the requirement of completeness and consistency. This is one of the best ways to improve product quality of safety-critical software and reduce costs of product development. Taking the avionics software system as an example, its corresponding civil airborne software airworthiness standard DO-178B/C [6] is to carry out various activities of avionics software research and development with multi-level requirements as the core. And it introduces the model-based system/software engineering methodology and formal methods [7], [8] to better support the achievement of safety and other related goals.

Gererally, the content of this article is arranged as follows. The second section gives a description of the framework for building a VRM model from natural language requirements. The third section defines the key concepts in the research framework. It includes the formal definition of the domain concept library, the design of the domain template library and the mathematical definition of its constituent elements, and the automatic construction algorithm of the table model in the VRM model. The fourth section is an example analysis of the Engine Indication and Crew Warning System (EICAS). The last section is the related work and summary.

II. Constructing VRM model for domain natural language requirements

This section first describes the framework of constructing formal requirement models(VRM)from domain natural language requirements. It includes domain natural language requirements template and automatic construction from standardized requirement models to VRM model. Then a brief description of the formal requirements model (called VRM model) is described.

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00017

A. Research Framework for VRM modeling

The work content of this article is part of the research content of an overall project ART (Avionics Requirement Tools) . The goal of the entire project is to design and implement a software tool platform (ART) for requirements analysis and verification in accordance with the requirements of DO-178B/C [9]–[12]. The first step is to build an engineering practical formal modeling method (VRM) that includes elements such as conditions, events, modes, and environmental interactions. In this process, it is necessary to consider the domain characteristics of modern civil aircraft avionics software requirements, natural language description of avionics domain requirements, and the "four-variable theoretical model" [13] formal methods. The second step is to conduct analysis and verification activities such as consistency and integrity based on the formal semantics of VRM model. Finally, the formalized VRM requirements model can be used to automatically generate test case sets.

The main work of this article is as follows. First, according to the characteristics of avionics area requirements and actual engineering requirements, a set of domain-oriented natural language requirement templates is designed and defined. Then, comprehensively consider the semantic elements of the adopted VRM model to form a requirement normalization method based on this template. Finally, an automatic construction algorithm from standardized requirement model to VRM model is given. The overall framework is shown in Figure 1.

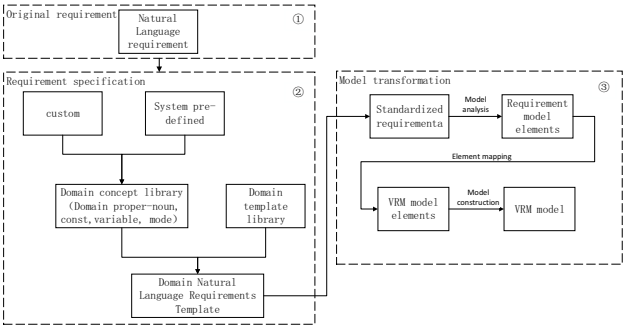


Fig. 1. Framework of generating VRM model based on domain natural language

Facing the domain natural language requirements, this article defines a set of domain natural language requirements templates. Then redefine the natural language requirements to form a standardized requirement model. The domain template is defined based on domain-related knowledge and actual requirements. It will describe data, proper-noun and fixed sentences contained in the requirements document in the form of templates. Eventually realize the unification of requirement description and reduce human error in the process of requirement definition. Its content includes: domain concept library and domain template library.

- The domain concept library includes: proper nouns, constants, input variables, output variables, term variables and mode class. In the domain concept library, proper nouns refer to proprietary system names, component names, etc. Constants refer to some predefined parameters in the software. Input variables refer to the data dynamically enter by a component during software operation. The output variable is the operation result of the component during the software operation. Term variables mean that some input data need to go through a series of operations to finally produce output data. In order to better describe this process, the VRM model need to add term variables to supplement the description. The mode class is the union of N non-empty disjoint modes. Each mode is an equivalent class of system state.
- According to the features of natural language requirements and the characteristics of VRM model elements, this paper defines four types of templates. After analyzing and summarizing the requirements example EICAS system, a conclusion is drawn: it is mainly divided into general functional requirements and display functional requirements. According to the characteristics of VRM model elements, conditional and event-based requirements are obtained. Therefore, a general conditional template, a general event type template, a display conditional template, and a display event type template are defined.
- Through domain natural language templates, we can redefine natural language requirements to obtain standardized requirements.

Based on the elemental characteristics of the standardized requirement model and VRM model, the data structure of the standardized requirement model and the VRM model is defined. VRM model can be automatically constructed by using VRM model construction algorithm.

- First, the regular expression of the domain template is defined. Then we can use this expression to parse the standardized requirement models to get the requirement model elements.
- According to the predefined requirement model element to VRM model element mapping table and domain template to VRM model table function mapping table, the requirement model element is mapped to VRM model element and VRM model table function information.
- According to the information of the VRM model elements and the VRM model table function, the VRM model can be automatically constructed.

B. VRM modeling

There are many formal methods and theories at present, but few methods are derived from actual engineering and are actually applied. And the description of most formal methods is too complicated, making it difficult for system engineers to understand and accept them. In fact, engineering personnel understand and are familiar with engineering domain knowledge and various tables (such as various avionics system operation manuals). Therefore, the VRM model used in this paper is a requirement model with both tabular and formal semantics.

Similar to the data storage and processing methods in relational database, the intuitive form of the VRM model is to use two-dimensional tables to build requirement models. The two-dimensional relational data in the relational database is actually a mathematical model strictly defined by the relational calculus logic system (ie: formal model). VRM model is based on a four-variable formal model and tailored according to the characteristics of the avionics software domain. Some of the formal definitions are as follows:

The six-tuple of the VRM requirement specification is {SV, C, E, F, TS, VR}. SV is the set of all state variables, which is a four-tuple, defined as: SV={MV, CV, M, IV }. It includes input variable MV, output variable CV, mode class M and term variable IV. The function of each data of the six-tuple is specifically described below.

- MV is a set of input variables that is non-empty and disjoint. MV={mv_1,mv_2,...,mv_l}. mv_1,mv_2,...,and mv_l are input variable.
- CV is a set of output variables that is non-empty and disjoint. CV={cv_1,cv_2,...,cv_l}. cv_1,cv_2,...,and cv_l are called output variables.
- M is a set of mode classes that is non-empty and disjoint. M={mc_1,mc_2,...,mc_m}. mc_1,mc_2,..., and mc_m are called mode class. And mc_i contains all modes in this mode class. Mci={mci_1,mci_2,...,mci_m}.
- IV is a set of term variables that is non-empty and disjoint. IV={iv_1,iv_2,...,iv_k}. iv_1,iv_2,..., and iv_k are called term variables.
- TS is the union of types. All types in TS are non-empty set of values.
- VR is a special function. It is used to map the name of the state variable to a specific value, indicating all the ranges of the state variable.
- C, as Condition, indicating a predicate on a single state variable. For example, $Altitude > 500$ means the current height is greater than 500. The condition is a logical expression which can be expressed in different ways. It can be a boolean variable which includes true and false, or a boolean expression $c_i \bigodot c_j$, etc. $\bigodot \in \{AND, OR, NOT\}$ represents logical operator. $C = r \circ v$. $\circ \in \{=, <, >, \neq, \geq, \leq\}$ represents the relational operator.
- E, as event, representing the predicate on two state variables. The general expression of the event is

$$EVENT(S) \textbf{ GUARD } D.$$

  $EVENT \in \{@T, @F, @C\}$ represents the event operator. $GUARD \in \{WHEN, WHERE, WHILE\}$ represents the guard operator.
- F is a table function. All tables are a mathematical function. It can be expressed by F_i.

The table functions in the VRM model include three categories: condition table, event table and mode transformation table. All three types of tables have corresponding formal semantic definitions. Due to space limitations, only a brief description of the examples in Figure 2 and Figure 3 is given. Figure 2 is an example of a condition table: based on the mode dependency sets D_n={Pressure, Overridden}. The value of the controlled variable SafetyInjection is defined (i.e. a functional requirement F1). The corresponding two-dimensional table visually represents the model and the corresponding mathematical logic formula model.

SafetyInjection=

F_1(Pressure, Overridden) = { Off if Pressure=high ∨ Pressure=Permitted ∨ (Pressure=TooLow ∧ Overridden=true); On if Pressure=TooLow ∧ Overridden=false }

Mode Pressure	**Condition**	
High, Permitted	True	False
TooLow	Overridden	Not Overridden
SafetyInjection	Off	On

Fig. 2. Condition table and its logic formula

Figure 3 is an example of an event table: based on the new and old mode dependency sets: {Block, Reset, Pressure, Overridden} and {Block, Reset, Pressure}. The function defines the value of the controlled variable Overridden (i.e. a functional requirement F2). Its corresponding two-dimensional table visually represents the model and the corresponding mathematical logic formula model.

Overridden' = F_2(Pressure, Block, Reset, Overridden, Pressure',Block',Reset') =

{ true if (Block'=On ∧ Block=Off ∧ Pressure=TooLow ∧ Reset=Off) ∨ (Block'=On ∧ Block=Off ∧ Pressure=Permitted ∧ Reset=Off)
false if (Reset'=On ∧ Reset=Off ∧ Pressure =TooLow) ∨ (Reset'=On ∧ Reset=Off ∧ Pressure =Permitted) ∨ (Pressure'=High ∧ Pressure≠High) ∨ ((Pressure'=Permitted ∨ Pressure'=TooLow) ∧ ¬(Pressure=Permitted ∨ Pressure=TooLow))
Overridden otherwise (no change) }

Mode Pressure	**Event**	
High	Never	@F(Pressure=High)
TooLow, Permitted	@T(Block=On) When Reset=Off	@T(Pressure=High) OR @T(Reset=On)
Overridden'=	true	false

Fig. 3. Event table and its logic formula

III. Key issues in the modeling framework

In this section, the related concepts involved in the research framework of Chapter 2 are defined in detail. First, the content of the domain concept library defined in the form of N-tuples is introduced. Second, it describes the domain template library and the mathematical definition of its constituent elements. Finally, the algorithm of automatic construction of VRM model is described.

A. Domain concept library

The domain concept library is built on specific domains and actual requirements. The main work is to collect the proper nouns, domain concepts and various data involved in the requirements, and describe their attributes in a format that conforms to formal semantics. The domain concept library contains the following contents: proper nouns, constants, variables, and mode class. The variables are divided into input variables, output variables, and term variables. And modes belong to the mode class. This article adopts the idea of object-oriented class definition to define the domain concept library and express it in the form of N-tuple.

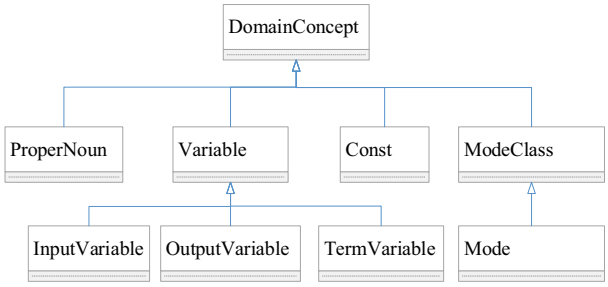


Fig. 4. Domain concept library hierarchy

The detailed definition of the domain concept library is shown in TABLEI.

Datatype contanied in the constant library and variable library includes two parts: system predefined and user-defined. The predefined parts of the system include Boolean, Char, Float, Double, Integer, String and Unsigned. The user-defined datatype is to redefine the range and precision on the basis of the systems predefined datatype, to obtain a new datatype that is more likely used in the actual project. The new datatype definition includes: the name, type, range and precision of the datatype. Redefinable datatype includes: Char, Integer, Float, Double, Unsigned, and Enumerated.

For example, the original requirement in the EICAS system, "When ipFADECEngineManualThrottleCmd is equal to TRUE, the color of the graphic symbol of the engine display thrust reference should be Green100.". It contains an input variable: ipFADECEngineManualThrottleCmd, an output variable: opFADECEngineThrustReferenceGraphicColor(from: the color of the graphic symbol of the engine display thrust reference) and a custom datatype: color(from: Green100). The detailed definition is shown in TABLEII.

B. Domain template library

The domain template library is a set of fixed sentences to describe the requirements. This article comprehensively considers the requirements description standards in the aviation industry, analyzes the requirements of the EICAS system, and divides the requirements into four basic types: general requirements, display requirements, functional requirements, and other requirements.

TABLE I
DEFINITION OF DOMAIN CONCEPT LIBRARY SEMANTICS

Name	Definition	Description
Domain-Concept	DomainConcept::= <Name, Description>	including the name and description of Domain-Cocept
Proper-Noun	ProperNoun::= <Name,Description>	including the name and description of ProperNoun
Constant	Constant::=<Name, Datatype, Value, Description>	including the name, Datatype, value and Description of Constant
Variable	Variable::=<Name, Datatype, Range, Accuracy, initial-Value, Description>	including the name, Datatype, Range, Accuracy, initialValue and Description of Variable
Input-Variable	InputVariable::= <Name, Datatype, Range, Accuracy, initialValue, Description>	including the name, Datatype, Range, Accuracy, initialValue and Description of InputVariable
Output-Variable	OutputVariabl::= <Name, Datatype, Range, Accuracy, initialValue, dependencyModeClass, Description>	including the name, Datatype, Range, Accuracy, initialValue, dependency-ModeClass, Description of OutputVariabl
Term-Variable	TermVariable::= <Name, Datatype, Range, Accuracy, initialValue, dependencyModeClass, Description>	including the name, Datatype, Range, Accuracy, initialValue, dependency-ModeClass, Description of TermVariabl
Mode-Class	ModeClass::= <Name, Description>	including the name and Description of ModeClass
Mode	Mode::=<Name, isInitial, isFinal, numericalValue, Description>	including the name, isInitial, isFinal, numericalValue and Description of Mode

TABLE II
DOMAIN CONCEPT LIBRARY OF REQUIREMENT EXAMPLE

Name	Definition
color	<color, Enumerated,(Green100),->
ipFADECEngine-ManualThrottle-Cmd	<ipFADECEngineManualThrottleCmd, boolean, (true,false), -. true>
opFADECEngine-ThrustReference-GraphicColor	<opFADECEngineThrustReference-GraphicColor, color, (Green100), -, Green100, ->

a) Domain template: According to the statistics of the requirement for EICAS instances, there are more general requirements and display requirements. According to the characteristic information of VRM model, it includes condition table and event table. Therefore, four basic domain templates were designed.

- General conditions: $< aircraft/system/equipment >$ shall be able to $< function > < object >$, when: $< condition >$.
- General events: $< aircraft/system/equipment >$ shall be able to $< function > < object >$, if: $< event >$.
- Display conditions: $< object >$ shall be able to

$< function > < format/requirement/standard >$, when: $< condition >$.

- Display event: $< object >$ shall be able to $< function > < format/requirement/standard >$, if: $< event >$.

Requirement templates are strictly defined using formal semantics. In the requirements template, the $< object > < airplane/system/equipment >$, $< function >$ and $< format/requirement/standard >$, which are domain concept, are defined in the chapter of domain concept library. $< condition >$ and $< event >$, which are condition and event, will be defined in the paragraph of condition template and event template.

The following content shows an example of the display condition template. As in the example in chapter of domain concept library, "opFADECEngineThrustReferenceGraphicColor" corresponds to $< object >$, "Green 100" corresponds to $< format/requirement/standard >$, "Display" corresponds to $< function >$, and "ipFADECEngineManualThrottleCmd=true" corresponds to $< condition >$.

b) Condition template: The condition refers to the value of the variable. Conditions may be compound. That means they are composed of simple conditions, and are connected by logical operators: AND, OR, and NOT.

The simple condition template uses triples as defined below:

$$ci = (O, R, V) \tag{1}$$

- O(Object): It usually is an input variable in the domain concept library.
- R(Relation): Relational operations, such as: $=, <, >, \geq, \leq$, etc.
- V (Value): Value refers to a specific value within the range of input variable.

Therefore, the condition template is defined using the following BNF paradigm. c represents a simple condition, C_term represents the items that make up the condition, and C represents the condition.

c ::= object '$>$' value | object '$<$' value | object '$=$' value | object '$\geq$' value | object '$\leq$' value

C_term ::= [NOT]c | C_term AND C_term | C_term OR C_term

C ::= C_term | C AND C_term | C OR C_term

For example, the condition of the EICAS requirement example given in chapter of domain concept library can be expressed as: ipFADECEngineManualThrottleCmd = true.

c) Event tempalte: In actual requirement, there is such a kind of requirement. It cannot be expressed by using simple conditions, so a new description mechanism needs to be introduced which is event. For example: when the height is lower than 500, the alarm is required to only alarm for 3 seconds, and then no longer alarm. If the conditional expression is used, an error will occur. And the alarm will always be in the alarm state, which has different semantics from the actual requirement. Therefore, it should be expressed in the form of event.

According to the definition of event expression, a total of 12 event templates can be summarized. So, the event templates is defined in TABLEIII.

TABLE III
EVENT TEMPLATES

Template	Semantic Definition	Description
@T(Ci)	NOT _Ci and Ci	Ci's last state is false, current state is true
@F(Ci)	_Ci and NOT Ci	Ci's last state is true, current state is false
@C(Ci)	NOT _Ci and Ci	Ci's last state is different from current state
@T(Ci) WHEN(Cj)	NOT _Ci and Ci and _Cj	Ci's last state is false, current state is true and Cj's last state is true
@F(Ci) WHEN(Cj)	_Ci and NOT Ci and _Cj	Ci's last state is true, current state is false and Cj's last state is true
@C(Ci) WHEN(Cj)	NOT _Ci and Ci and _Cj	Ci's last state is different from current state and Cj's last state is true
@T(Ci) WHILE(Cj)	NOT _Ci and Ci and Cj	Ci's last state is false ,current state is true and Cj's state is true
@F(Ci) WHILE(Cj)	_Ci and NOT Ci and Cj	Ci's last state is true, current state is false and Cj's state is true
@C(Ci) WHILE(Cj)	NOT _Ci and Ci and Cj	Ci's last state is different from current state and Cj's state is true
@T(Ci) WHERE(Cj)	NOT _Ci and Ci and _Cj and Cj	Ci's last state is false, current state is true, Cj's last state and current state are both true
@F(Ci) WHERE(Cj)	_Ci and NOT Ci and _Cj and Cj	Ci's last state is true, current state is false, Cj's last state and current state are both true
@C(Ci) WHERE(Cj)	NOT _Ci and Ci and _Cj and Cj	Ci's last state is different from current state, Cj's last state and current state are both true

C. Model construction

The process from the standardized requirement model to the automatic construction of the VRM model mainly includes two parts.

- Conversion of the domain concept library: The domain concept library is converted into the constant dictionary, variable dictionary and user-defined datatype in the VRM model. There is a one-to-one correspondence between the domain concept library and the data dictionary.
- Conversion of standardized requirement: The semantics of standardized requirements is, "Under a certain condition or event, a variable (a variable is an output variable or a term variable) obtains a specific value within its range." The range of output variable and the term variable should contain at least one value, so the domain natural language requirement and table function in VRM model have a many-to-one conversion relationship.

a) Standardized requirement model: Based on the contents defined in the above two subsections, the data structure of the standardized requirement model is shown in Figure 5.

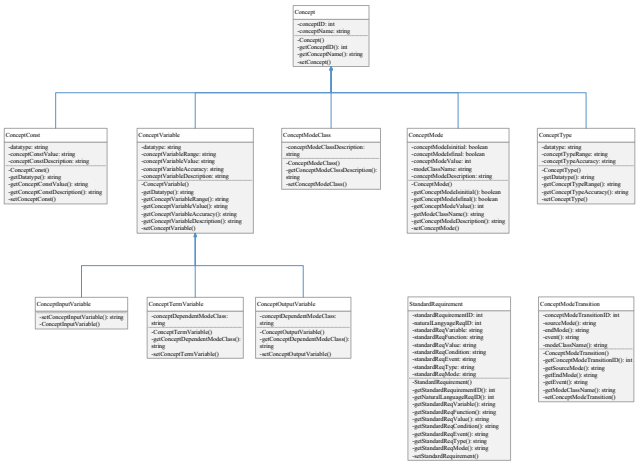


Fig. 5. Standardized requirement model

The standardized requirement model is mainly divided into two parts: domain concept library and standardized requirements. Domain concept libraries include constant concept libraries, custom datatype concept libraries, input variable concept libraries, term variable concept libraries, output variable concept libraries, mode class concept libraries and mode transition concept libraries. Standardization requirements include general condition normalization requirements, general event normalization requirements, display condition normalization requirements, and display event normalization requirements.

b) VRM model: The VRM model includes constant dictionaries, custom datatype, variable dictionaries, and behavior tables. In order to better express the model information and the specific implementation of the subsequent tools, its content has been integrated and converted into a constant dictionary, custom datatype, input variable dictionary, output variable (term variable) condition table, and output variable (term variable) Event table and mode conversion table. In summary, the data structure of the VRM model is shown in Figure 6.

c) VRM model construction: There are three types of VRM model construction.

Construction of constant dictionaries, custom datatype, and input variable dictionaries. Taking the constant dictionary as an example, the construction algorithm is shown in Figure 7.

The input of the constant dictionary construction algorithm is the domain concept of constant, and the output is constant dictionary. A single constant is stored using an instantiated object of the constant dictionary class, and the constant dictionary is an array of constant objects. The algorithm complexity is O(n).

The behavior table includes output variable (term variable) condition table and output variable (term variable) event table. The construction algorithm is shown in Figure 8.

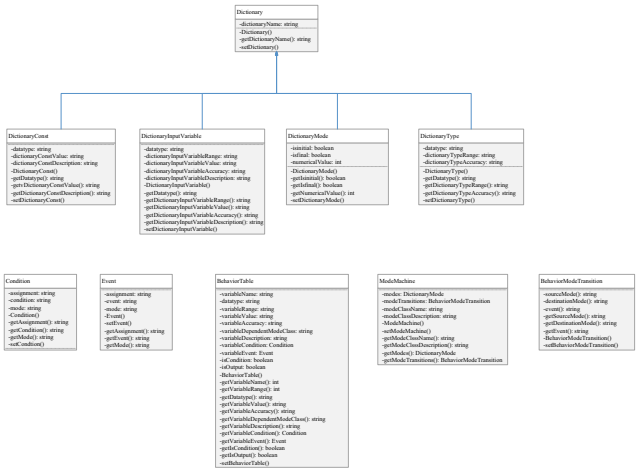


Fig. 6. VRM model

Input: ConceptConst
Output: DictionaryConst
1: DictionaryConst dc = new DictionaryConst();
2: ArrayList <DictionaryConst> arrayDC = new ArrayList <DictionaryConst>();
3: **for** each ConceptConst cc in arrayConceptConst **do**
4: DictionaryConstDataValue = cc.getValue();
5: dc.setDictionaryConst();
6: arrayDC.add(dc);
7: **end for**

Fig. 7. Constant dictionary construction algorithm

The input of the behavior table construction algorithm is the domain concept of the term variable, the domain concept of output variable and the standardized requirements. And the output is the VRM behavior table. The term variable domain concept and the output variable domain concept are combined and stored as the variable domain concept. The behavior table mainly contains two aspects: the domain concept content of the variable and the behavior information (conditions and

Input:
ConceptTermVariable、ConceptOutputVariable、StandardRequirement
Output:BehaviorTable
1: Condition c = new Condition();
2: ArrayList <Condition> arrayC = new ArrayList <Condition>()
3: Event e = new Event();
4: ArrayList <Event> arrayE = new ArrayList <Event>();
5: BehaviorTable bt = new Behavior();
6: ArrayList <BehaviorTable> arrayBT = new ArrayList <BehaviorTable>();
7: **for** each ConceptVariable cv in arrayConceptVariable **do**
8: DictionaryVariableDataValue = cv.getValue();
9: **for** each StandardRequirement sr in arraySR **do**
10: **if** sr.getVariableName().equals(cv.getName()) **then**
11: **if** sr.getReqType.euqals("Condition") **do**
12: ConditionDataValue = sr.getValue();
13: c.setCondition();
14: arrayC.add(c);
15: **end if**
16: **if** sr.getReqType.equals("Event") **do**
17: EventDataValue = sr.getValue();
18: e.setEvent();
19: arrayE.add(e);
20: **end if**
21: **end if**
22: **end for**
23: bt.setBehaviorTable();
24: arrayBT.add(bt);
25: arrayC = new ArrayList <Condition>();
26: arrayE = new ArrayList <Event>();
27: **end for**

Fig. 8. Behavior table construction algorithm

events) contained in the standardized requirements. Therefore, an array of condition objects, an array of event objects, and an array of behavior table objects are defined. Algorithm complexity is O(n2).

The construction algorithm of the mode transition table is shown in Figure 9.

```
Input: ConceptModeClass、ConceptMode、ConceptModeTransition
Output:ModeMachine
1: Mode m = new Mode();
2: ArrayList <Mode> arrayM = new ArrayList  <Mode>()
3: BehaviorModeTransition bmt = new BehaviorModeTransition();
4: ArrayList < BehaviorModeTransition > arrayBMT = new ArrayList
< BehaviorModeTransition >();
5: ModeMachine mm = new ModeMachine();
6: ArrayList <ModeMachine> arrayMM = new ArrayList
 <ModeMachine>();
7: for each ConceptModeClass mc in arrayMC do
8:   ModeClassDataValue = mc.getValue();
9:   for each ConceptMode cm in arrayCM do
10:  if cm.getModeClassName.equals(mc.getName()) then
11:      ModeDataValue = cm.getValue();
12:      m.setMode();
13:    arrayM.add(m);
14:  end if
15:  end for
16:  for each ConceptModeTransition cmt in arrayCMT do
17:  if cmt.getModeClassName.equals(mc.getName()) then
18:    BehaviorModeTransitionDataValue = cmt.getValue();
19:    bmt.setBehaviorModeTransition();
20:    arrayBMT.add(bmt);
21:  end if
22:  end for
23:  mm.setModeMachine();
24:  arrayMM.add(mm);
25:  arrayM = new arrayList <Mode>();
26:  arrayBMT = new arrayList<BehaviorModeTransition>
27:  end for
```

Fig. 9. Mode transition table construction algorithm

The input of mode transition table construction algorithm is the domain concept of mode class and the domain concept of mode transition. And the output is VRM mode transition table. The mode transition table contains the mode class information, the modes contained in the mode class and the mode conversion information under related events. Therefore, a mode object array and a mode transition object array are defined. The algorithm complexity is O(n2).

IV. EICAS CASE ANALYSIS

A. EICAS system overview

Engine Indicating and Crew Alerting System (EICAS) is used to indicate the working status of various aircraft systems. It provides text, graphics and audio information, and prompt faults and issue warnings when faults occur. EICAS is divided into two areas: engine instructions and crew warning messages. The engine indication area contains engine parameters and flap position, fuel quantity, and landing gear position information. The unit warning information is displayed in the upper right corner of EICAS, which is used to prompt warning information and warning information. The interface diagram information is shown in Figure 10.

B. EICAS VRM model generation

In order to carry out the case analysis of the requirement model establishment and VRM model construction , some typical EICAS actual requirements are selected as follows.

- The Engine HF shall determine ipFlightDeckUnitsConfigurationIsMetric to be METRIC when (the parameter ipFlightDeckUnitsConfigurationIsMetric is valid and equal to TRUE) otherwise as IMPERIAL.
- The color of the Engine Display Thrust Reference graphical symbol shall be Green 100 when ipFADECEngineManualThrottleCmd is equal to TRUE.
- The Engine Display shall display the command sector in Yellow 50 color if the associated ipFADECEngineThrust[L|R] is moving away from the associated ipFADECEngineThrustCommand[L|R].

Fig. 10. EICAS display information concept graphic

By constructing the domain concept library for the above three original requirements, the following content is obtained(i.e. TABLEIV-TABLEVII).

TABLE IV
INPUT VARIABLE DOMAIN CONCEPT

No.	Name	Data-type	Range	Accu-racy	Initial-Value	Descri-ption
1	ipFDUCon-figuration-IsMertic-Status	value-Status	(in-valid, valid)	-	invalid	-
2	ipFADECEn-gineManual-ThrottleCmd	bool-ean	(true, false)	-	false	-
3	ipFADECEn-gineThrust-[L—R]	float	(-1* E+5,1* E+5)	-	-	-
4	ipFADECEn-gineThrust-Command-[L—R]	float	(-1* E+5,1* E+5)	-	-	-
5	ipFDUConfi-gurationIs-Metric	bool-ean	(true, false)	-	false	-

According to the domain template library defined in the previous chapter and the domain concept library defined above, the original requirements are redefined as the following standardized requirements.

- opFDUConfigurationFormat shall be able to set to METRIC, when : $ipFDUConfigurationIsMetric = true$ and $ipFDUConfigurationIsMetricStatus = valid$.
- opFDUConfigurationFormat shall be able to set to IMPERIAL, when : $ipFDUConfigurationIsMetric = false$ and $ipFDUConfigurationIsMetricStatus = valid$.
- opFDUConfigurationFormat shall be able to set to IMPERIAL, when : $ipFDUConfigurationIsMetric = false$ and $ipFDUConfigurationIsMetricStatus = invalid$.
- opFDUConfigurationFormat shall be able to set to METRIC, when :

TABLE V
OUTPUT VARIABLE DOMAIN CONCEPT

No.	Name	Data-type	Range	Accu-racy	Ini-tial-Value	Mod-eCl-ass	Des-crip-tion
1	opFDUC-onfigu-ration-Format	FDUP-aram-eter-Form-at	(imp-eria-l,me-tric)	-	metr-ic	def-ault	-
2	opFADE-CEngin-eThrus-tRefer-enceGr-aphic-Color	color	(gre-en100 ,yel-low50)	-	gree-n100	def-ault	-
3	opN1Co-mmandS-ector-Color	color	(gre-en100 ,yel-low50)	-	gree-n100	def-ault	-

TABLE VI
TERM VARIABLE DOMAIN CONCEPT

No.	Name	Data-type	Range	Accu-racy	Ini-tial-Value	Mod-eCl-ass	Des-crip-tion
1	tAbsol-uteVal-ueComm-andAnd-Thrust	color	(gre-en100 ,yel-low50)	-	gree-n100	def-ault	-

$ipFDUConfigurationIsMetric = true$ and $ipFDUConfigurationIsMetricStatus = invalid$.

- opFADECEngineThrustReferenceGraphicColor shall be able to display as Green 100, when : $ipFADECEngineManualThrottleCmd = true$.
- opN1CommandSectorColor shall be able to display as yellow 50, if : $@C(tAbsoluteValueCommandAndThrust)$ when $(tAbsoluteValueCommandAndThrust > 0)$

The VRM models are automatically constructed through the construction algorithm of VRM model. Its XML document is shown in Figure 11.

The standardized requirement model constructed with tools is shown in the Figure 12 and Figure 13.

TABLE VII
CUSTOM DATATYPE

No.	Name	Data-type	Range	Accuracy
1	value-Status	Enumer-ated	(invalid, valid)	-
2	FDUPar-meter-Format	Enumer-ated	(imperial, metric)	-
3	color	Enumer-ated	(green100, yellow50)	-

```
<?xml version="1.0" encoding="UTF-8"?>

<VRMmodel>
<requirements>
    <requirement>
    <requirement>
    <requirement>
</requirements>

<standardRequirements>
    <standardRequirement type="genralConditon">
    <standardRequirement type="genralConditon">
    <standardRequirement type="genralConditon">
    <standardRequirement type="genralConditon">
    <standardRequirement type="displayConditon">
    <standardRequirement type="displayEvent">
</standardRequirements>
<constants/>
<types>
    <type name="FDUParameterFormat">
    <type name="valueStauts">
    <type name="color">
</types>
<inputs>

<tables>
    <table name="opFADECEngineThrustReferenceGraphicColor" isOutput="1">
    <table name="tAbsoluteValueCommandAndThrust" isOutput="0">
    <table name="opN1CommandSectorColor" isOutput="1">
</tables>
<stateMachines/>
</VRMmodel>
```

Fig. 11. The XML document of VRM model

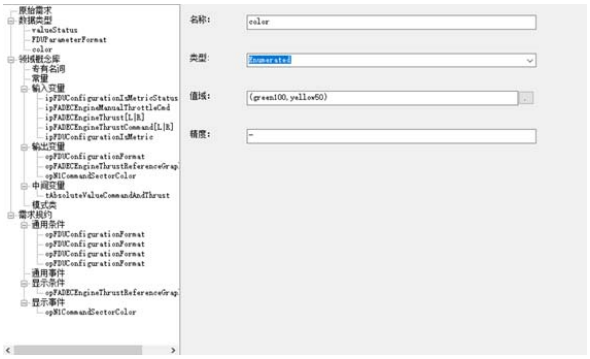


Fig. 12. Domain concept library and standardized requirement model

Fig. 13. VRM model

V. Related work and summary

At present, in the area of avionics, from the perspective of software requirements modeling, related work is roughly divided into the following categories. The first is the theory and technology formed from the development engineering experience of actual safety-critical systems, such as: four-variable model [14], SCR method [15], RSML method [16], CoRE method [17], SpecTRM [18] etc. The second is the software requirements specification method generated from the area of software engineering, such as: the requirements capture and description method of the Use-Case model [20] in the Unified Modeling Language (UML) [19], and the parameter model used to describe the system requirements in the System Modeling Language (SysML). Their typical tools include: Raphsody, Statmate [21], etc. The third is the requirement modeling and code generation technology developed from the synchronous data flow language of electronic hardware system design, such as: the Simulink tool [22], SCADE tools based on Esterel technology. The last category is to describe the system and software requirements in natural language with a limited structure [23].

This paper proposes a templated specification method for domain natural language requirements. The domain concept libraries and domain template libraries that strictly meet mathematical semantics are defined. In order to reduce ambiguity of requirement, the natural language requirement is redefined based on the template. Secondly, the automatic conversion rules from the standardized requirement model to the VRM model are given. It includes conversion rules from domain concept library to data dictionary and conversion rules from domain template library to behavior table. Finally, a prototype tool was implemented on the .net platform, and a case analysis was carried out based on the EICAS system.

References

[1] Leveson N. Engineering a safer world : system thinking applied to safety[M], MIT Press, 2011.

[2] Heeager, L.T, Nielsen, P.A. Agility in Development of Safety-Critical Software : A Conceptual Model[J]. Information and Software Technology, 2018, v103: 22-39.

[3] Rierson L. Developing safety-critical software : a practical guide for aviation software and DO-178C compliance[M]. CRC Press, 2013.

[4] Keith C. Report on the serious incident to Boeing B787-8, ET-AOP London Heathrow Airport 12 July 2013[M]. AAIB Press, 2013.

[5] ASN. Accident list: Boeing 787[DB/OL]. http://aviation-safety.net/database/types/Boeing-787/database, 2020.

[6] Stephen A., Jacklin. Certification of Safety-Critical Software Under DO-178C and DO-278A[C]. United States : Infotech@Aerospace 2012, 2012: 15-28.

[7] Liu S.Y. Agile formal engineering method for software productivity and reliability[C]. Russia : Association for Computing Machinery, 2018: 1-6.

[8] Jharko E.P. The methodology of software quality assurance for safety-critical systems[C]. Russia: International Siberian Conference on Control and Communications, 2015:1-5.

[9] RTCA DO-331. Model-Based Development and Verification Supplement to DO-178C and DO-278A[M]. RTCA Press, 2011.

[10] RTCA DO-332. Object-Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A[M]. RTCA Press, 2011.

[11] RTCA DO-333. Formal Methods Supplement to DO-178C and DO-278A[M]. RTCA Press, 2011.

[12] Patcas, Lucian M. Implementability of requirements in the four-variable model[C]. Canada: Science of Computer Programming, 2015: 339-362.

[13] Buckl, Christian. Model-Based analysis and development of dependable systems[C]. Germany: Lecture Notes in Computer Science, 2010: 271-293.

[14] Patacas L.M, Lawford M, Maibaum T. From System Requirements to Software Requirements in the Four-Variable Model[J/OL]. http://core.al.uk/display/23645454.

[15] Hager J A. Software cost reduction methods in practice[J]. IEEE Transactions on software engineering, 1989, 15(12):1638-1644.

[16] Lutz R R. Analyzing software errors in safety-critical embedded system[C]. Proc of the 1st IEEE International Symposium on Requirements Engineering, 1993:126-133.

[17] Perseil I, Pautet L. Formal methods integration in software engineering[J]. Innovations in Systems and Software En-gineering, 2010, 6(1): 5-11.

[18] Fan, Chin F, Cheng C.Y. Constraint-based software specifi-cations and verification using UML[J]. IEICE Transactions on Information and Systems, 2006, E89-D(6): 1914-1921.

[19] Huning, Lars, Iyenghar. A UML profile for automatic code generation of optimistic graceful degradation features at the application level[C]. MODELSWARD 2020 – Pr-oceedings of the 8th International Conference on Model-Driven Engineering and Software Development, 2020: 336-343.

[20] Yue T, Briand L.C, Labiche Y. Facilitating the transition from use case models to analysis models[J]. ACM Transitions on Software Engineering and Methodology, 2013, 22(1): 1-38.

[21] Van M, Simon, Vangheluwe, Hans. Introduction to state-charts modeling, simulation, testing and deployment[C]. Pro-ceedings – Winter Simulation Conference, 2018: 306-320.

[22] Biswas D., Seth S., Bor M. A Study of the Dynamics of a New Piecewise Smooth Map[J]. International Journal of Bifurcation and Chaos, 2020, v30(1).

[23] Zhang H.H, Yue T, Ali S. A Restricted Natural Language Based Use Case Modeling Methodology for Real-Time Sys-tems[C]. Proceedings – 2017 IEEE/ACM 9th International Workshop on Modelling in Software Engineering, 2017: 5-11.

RTI-Grain: A Method for Detecting the Foreign Body of Granary Based on RSS

Chunhua Zhu*
Key Laboratory of Grain Information Processing and Control
College of Information Science and Engineering, Henan University of Technology,
Zhengzhou, Henan, China, 450001
e-mail: zhuchunhua@haut.edu.cn

Jiake Tian
Key Laboratory of Grain Information Processing and Control
College of Information Science and Engineering, Henan University of Technology,
Zhengzhou, Henan, China, 450001
e-mail: 2581423854@qq.com

Zhen Shi
Key Laboratory of Grain Information Processing and Control
College of Information Science and Engineering, Henan University of Technology,
Zhengzhou, Henan, China, 450001
e-mail: 1060257290@qq.com

Jing Yang
Key Laboratory of Grain Information Processing and Control
College of Information Science and Engineering, Henan University of Technology,
Zhengzhou, Henan, China, 450001
e-mail: yangjing@haut.edu.cn

***Abstract*—This paper proposes a non-destructive, fast, low-cost, non-contact method for detecting stored grain foreign body based on received signal strength (RSS) information , RTI-Grain. The feasibility of stored grain foreign body detection was verified by using RSS data. On this basis, the RTI-Grain detection method was designed. Firstly, the RSSI data is subjected to data normalization, noise elimination and averaging preprocessing, and then ellipse weight model was used to compute the attenuation coefficients in each pixel, finally the Tikhonov regularization was adopted to reconstruct the image of the abnormal area. The experimental verification shows that the foreign body of empty hole, metal and water in granary can be detected, and the detection accuracy is 98.1% at maximum.**

Keywords-component; stored grain; received signal strength information (RSSI); radio tomography imaging(RTI); image reconstruction

I. INTRODUCTION

Grain safety has always been a serious problem in the world. In the process of grain storage, transportation and processing, in order to avoid fraud and other irregularities which have occurred in the occurred in the management, the country will make an annual clearance investigation on the granary, which requires a lot of manpower, material and financial resources [1], therefore, it is urgent to carry out research work granary reserves lossless verification. Existing lossless foreign body detection adopt the wired sensor network [2] and electromagnetic wave based on ground penetrating radar(GPR)[3], the wired sensor network detection need to bury the wired sensors in the granary, which makes it difficult to layout and generally be used to detect the temperature and humidity; the GPR method can effectively detect the foreign body with lower frequency for guaranteeing the penetrating depth, because the transmitting waves with the lower frequency will bring the more powerful echo waves, thereby the imaging the foreign body can be completed. Therefore, the penetrating resolution is not too high and expensive for the GPR method. However, With the development of wireless sensor networks (WSNs), passive target detection technology has become a hot issue in the field of positioning. In recent years, radio tomography imaging (RTI) technology based on WSNs has been developed, in which the received signal strength (RSS) measurements among the static wireless sensors is used to image the changes in the radio propagation environment in the areas of the sensors. As a result, the changes caused by foreign body can be detected and tracked, which have been verified for human body positioning in indoor and outdoor environments with lower power and less cost [5,6]. In an RTI system, accurate localization depends on an accurate model for RSS measurements, and the elliptical model is a typical one to model RSS and to perform localization. In this paper, we presented one new foreign body detection based on received signal strength information (RSSI), RTI-Grain. For grain media, the wireless signal attenuation characteristics are different with indoor or outdoor environments, which brings new problems in RTI-Grain method.

The main contributions of this paper are as follows.

- Verify the feasibility of using RTI technology to detect foreign body of granary by comparing the RSSI data under normal grain and abnormal grain conditions.
- Design RTI-Grain penetrating system, including RSSI data acquisition experiment platform and foreign body image reconstructing model.
- Conduct an experiment to verify the performance of RTI-Grain detection, where three foreign bodies of empty hole, metal and water in granary are used.

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00018

The rest of the paper is organized as follows: Section 2 discusses the concept and application of RTI, While Section 3 introduces the feasibility study of RTI-Grain foreign body detection. In Section 4, We introduces RTI-Grain system model. Section 5 verifies the performance of RTI-Grain detection method, and the conclusion is presented in Section 6.

II. PRELIMINARIES

In this section, we first introduce the basic concept of RTI technology, and then introduce the extensive application of RTI technology.

A. RTI concept

Radio tomography imaging(RTI) [4-7] technology was proposed by Joey Wilson and Neal Patwari of Utah university, as shown in Fig. 1. From Fig. 1, there is one monitored area by several sensor nodes, when the target enters the monitored area, target will reflect or keep out the wireless signal between nodes, thereby, the RSSI of each node can embody the shading degree and whether or not there is one target emerging in the monitored area, even the target position can be obtained by measuring the shadow attenuation between the nodes of wireless sensor network, as shown in Fig. 2.

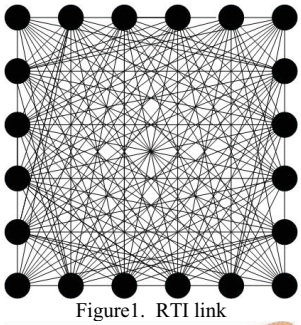


Figure1. RTI link

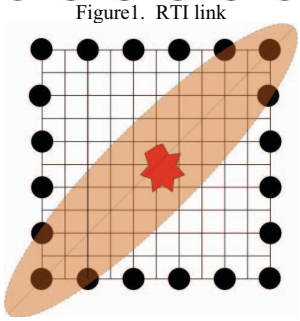


Figure 2. RTI network

Here, the RTI technology does not require the target to carry an electronic tag or sensor node, which is called the passive positioning.

B. RTI application

RTI technology is widely used with the development of wireless sensor network(WSN). For example, Alippi [8] applied RTI technology for outdoor large-scale target detection and localization, here 20 nodes are used to cover the forest environment with the size of 35m × 60m and the localization error of 3.2m is obtained. Anderson [9] applied RTI technology for vehicle identification and location on the road, and can reconstruct the image of the cross section and longitudinal section of the vehicle through 3D RTI technology. Mostofi [10] installed a pair of nodes on two mobile robots, which can scan and image the closed area to obtain the internal structure information of the closed area. Bocca [11] realized the localization of family members in the family environment through RTI technology, which can be used to monitor the elderly or patients in the family. At present, RTI technology has been applied in target locating and tracking [12,13], vehicle identification [14], wall-penetrating detection [15,16], medical and health care [17] and other fields.

III. FEASIBILITY OF DETECTING FOREIGN BODY BASED ON RTI

RTI technology has not been used in target positioning of grain medium, in grain granary the signal transmission has the characteristics of scattering and serious attenuation. Here the sensibility of RSSI on emerging target and the deployment of WSN nodes all can be one new challenge. In order to verify the feasibility of using RSS data to detect foreign body in grain medium, the following experiments will be carried out.

A. Experiment 1

Assuming two cases: one is that the monitored area is normal grain granary, and the other is abnormal grain granary of existing foreign body, here using one glassware of diameter 8CM as the foreign body, As shown in Fig. 3, when existing foreign body in grain medium, the RSSI of WSN nodes has shown significant difference, the existing foreign body can brought the change of RSS data, thereby, RSS-based detection of foreign body in grain medium is of more feasibility.

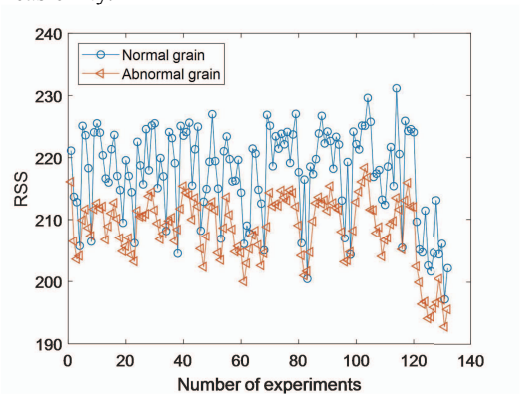
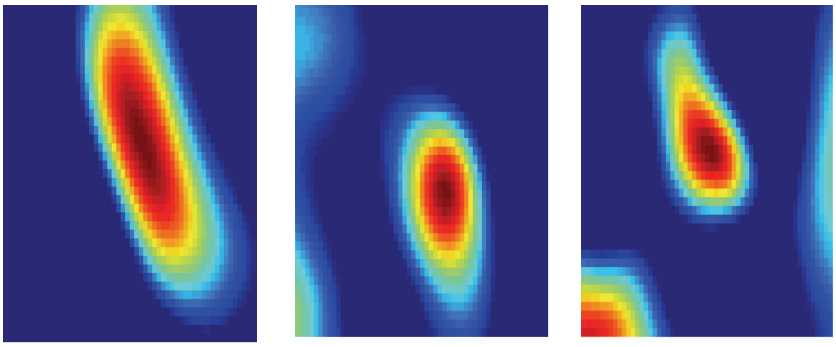


Figure3. The RSSI under abnormal grain condition



(a) Four nodes (b) Six nodes (c) Eight nodes

Figure4. Image reconstruction under different WSN nodes

B. *Experiment 2*

On the basis of Experiment 1, we continue to explore whether RTI technology can locate the foreign body by image reconstruction of the monitored area. In experiment, four WSN nodes are deployed around the simulated granary, the reconstructed monitored area image is shown in Fig. 4(a), and Fig. 4(b) and Fig. 4(c) is for six WSN nodes and eight WSN nodes. Despite of the fewer WSN nodes, it is feasible to position foreign body. Here we can optimize the deployment of WSN nodes to improve the positioning accuracy.

IV. RTI-BASED ABNORMAL GRAIN CONDITION DETECTION

A. *System model*

As shown in Fig. 5, the RTI-based abnormal grain condition detection system proposed in this paper mainly works in two stages: offline training stage and online detection stage.

- Offline training stage: several WSN nodes are arranged around the simulated grain granary with normal grain condition. After the success of networking, the WSN network will automatically send and receive wireless signals for ten minutes, the received RSS data will be trained and extracted the RSS characteristics of each sensor node, and there by a fingerprint database with normal grain condition will be established.
- Online detection stage: for the grain granary with abnormal grain condition, the RSSI data from WSN nodes will be collected and compared with that of fingerprint database, and the difference between them will be computed for the following foreign body analysis.

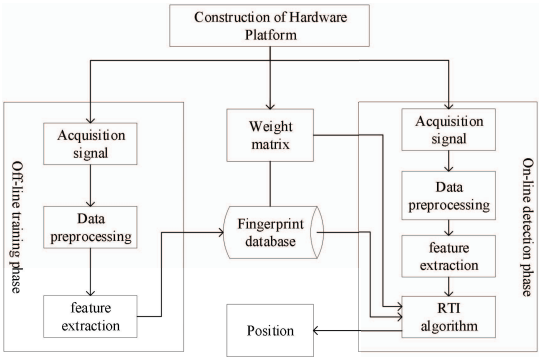


Figure 5. RTI-Grain abnormal Grain condition detection system

The core part of the system is RSS data collection and data processing. Data acquisition is shown in Fig. 6. here, there are a number of sensor ball nodes with CC2530 RF module driven by STC12C5A60S2 module for RSS data collection, and Dell laptop for recording RSS data of all WSN nodes and generating one TXT file. In Fig. 6, the locating process includes three modules: data preprocessing, feature extraction and RTI image reconstruction. Data preprocessing can eliminate the noise and multipath interference of the RSS data and normalize the data, then the RSSI change will be extracted as data features, finally, RTI technology can find the location of the foreign body by RSS features.

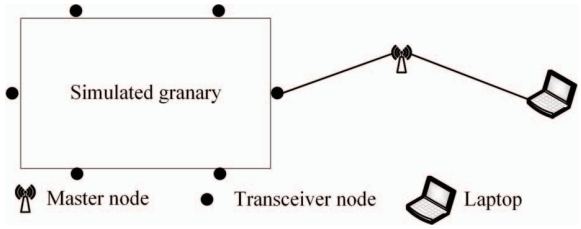


Figure 6. RSS data collection

B. *Data preprocessing*

We calibrate RSS data through gaussian low-pass filtering and data normalization to reduce the impact of noise.

The grain belongs to the granular body. When the wireless signal is transmitted in the granary, the grain surface is easy to reflect. Because the RSS change caused

by the grain reflection belongs to the fast fading, it is usually modeled as noise. When the system processes the signal in time domain, it uses low-pass filter to reduce the influence of fast fading. The low-pass filter uses one-dimensional Gaussian low-pass filter [18,19].Suppose ω is the size of Gaussian filtering window, δ is the standard deviation of Gaussian function, and output signal can be expressed as

$$x_{l,t} = \sum_{t-\omega}^{t+\omega} f(i) * x_{l,i} \quad (1)$$

Among $f(i) = \exp(-(i-r)^2 / (2*\delta^2)) / \sqrt{2*\delta^2}$ is a Gaussian mask and $x_{l,t-\omega} \cdots x_{l,i} \cdots x_{l,t+\omega}$ is the RSS value of the *l*-th link. It should be noted that the filtering window ω will increase the delay too much, and the drying effect will be poor if it is too small. After weighing the delay and effectiveness, the window size is selected based on experience.

In order to further reduce the impact of noise, improve the detection accuracy and reconstruction location accuracy. We normalize the filtered data [20]. The normalized values are expressed as:

$$r_{l,t} = \frac{x_{l,t} - x_{mean}}{x_{\max} - x_{\min}} \quad (2)$$

Among them, $x_{l,t}$ represents the original data, x_{mean} represents the mean value of RSS data, $x_{\max}$ and $x_{\min}$ represent the maximum and minimum value of RSS in the current time period respectively.

C. Feature extraction

The extracted features are mean or variance features. Each sensor node collects RSS data at different times, and obtains normalized data matrix through data preprocessing. The mean value of normalized RSS matrix is expressed as

$$\overline{r_l} = \frac{\sum r_{l,t}}{N_l} \quad (3)$$

Here N_l represents the number of data on the *l*-th link, which is determined by the length of RSS data.

D. RTI-based location

As shown in Fig. 7, the RTI include three key techniques, that is Communication link loss model, ellipse weight model and image reconstruction. The communication link loss model is usually linear, which can establish the relationship between RSS attenuation and grid pixels; elliptical weight model of RTI system can describe the weighting values of different pixels affected by target shading; and target location can be reconstructed by solving conditioned equations. .

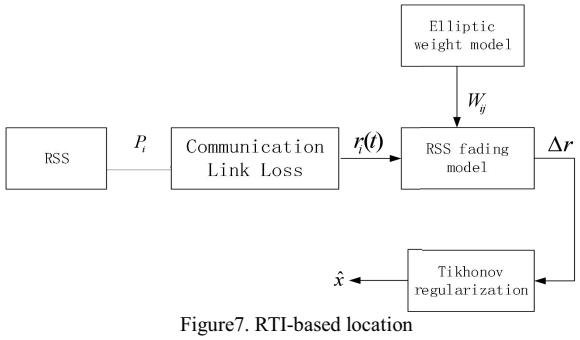


Figure7. RTI-based location

1) Communication link loss model

When there is no target in the monitoring area, the wireless signal satisfies the path loss model [21], and the mean value of RSS measurement on the *l*-th link is recorded as $\overline{r_l}$. When the target enters the monitoring area, the RSS measurement at the time *t* of the *l*-th link is recorded as $r_{l,t}$, the RSS change generated by the target is expressed as

$$\Delta r_{l,t} = \overline{r_l} - r_{l,t} = S_{l,t} + n_{l,t} \quad (4)$$

Where

$$\overline{r_l} = P_s - L_l$$

$$r_{l,t} = P_s - L_l - S_{l,t} - n_{l,t}$$

Here $\Delta r_{l,t}$ represents the RSS change caused by the target, $S_{l,t}$ represents the shadow fading caused by the target occlusion link, $n_{l,t}$ represents the multipath effect and the impact of noise, P_s represents the transmission power of the tactile sensor node, and L_l represents the path loss.

As shown in Fig. 2, the monitoring area is evenly divided into *N* pixels. The shadow fading caused by the target can be expressed as the sum of the attenuation weighting of each pixel, and the RSS change on the *l*-th link can be expressed as

$$\Delta r_{l,t} = \sum_{j=1}^{N} w_{lj} \Delta x_{j,t} + n_{l,t} \quad (5)$$

Here $\Delta x_{j,t}$ represents the RSS change of the monitoring area in the j-th grid, and $w_{l,}$ represents the weight of the *j*-th grid on the *l*-th link.

Considering both *N* grids and *L* communication links, *L* linear equations can be obtained, which are expressed in the form of matrix.

$$\mathbf{\Delta r_t} = \mathbf{W \Delta x_t} + \mathbf{n_t} \quad (6)$$

Among $\mathbf{\Delta r_t}$ is the measurement matrix composed of the RSS changes of all links at time *t,* $\mathbf{\Delta x_t}$ is the RSS shadow fading vector of all grids at time *t*, and $\mathbf{n_t}$ is the noise vector. $\mathbf{W}$ is the vector matrix of *N* grid weights on each link.

2) Ellipse weight model

This section adopts the traditional ellipse weight model [5]. The area affected by the target is modeled as an ellipse with the transmitter and receiver node as the focus, as

shown in Fig. 2. The weight of the grid in the ellipse that has an impact on the RSS change is 1, the weight of the grid outside the ellipse that has no impact on the RSS change is 0, and the weight network is expressed as

$$w_{lj} = \begin{cases} 1 & d_{lj}(1) + d_{lj}(2) < d_l + \lambda \\ 0 & otherwise \end{cases} \quad (7)$$

Among them, d_l is the distance between the transmitter and receiver node of link L, $d_{ij}(1)$ and $d_{lj}(2)$ are the distance between grid J and two sensors, λ is the parameter to adjust the elliptical range.

3) Image reconstruction

After the elliptic weight matrix is determined, the linear equations can be solved to find the target location by imaging. The number of unknown numbers n in the equations is far greater than the number L of the equations, so the regularization method needs to be used. Tikhonov regularization [22-23] is an effective method for solving ill conditioned equations. First, we minimize the objective function.

$$\min_{\Delta \mathbf{x}_t} \left\| \Delta \mathbf{r}_t - \mathbf{W} \Delta \mathbf{x}_t \right\|^2 + \alpha \left\| \Gamma \Delta \mathbf{x}_t \right\|^2 \quad (8)$$

Here, Γ is Tikhonov matrix, which represents the prior information of model solution, $\left\| \Gamma \Delta \mathbf{x}_t \right\|$ Represents the punishment term of Tikhonov regularization, α is an adjustable regularization parameter, its size determines whether the final solution is biased to measurement data or to prior information. Then the approximate estimation of Δx_t can be obtained by deriving equation (9), as follows

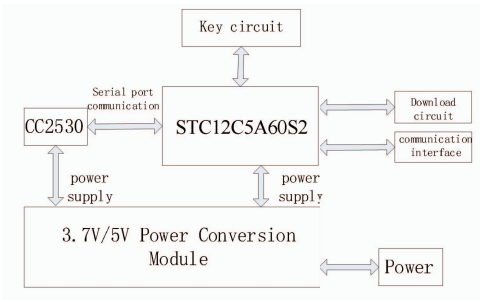
$$\Delta \hat{\mathbf{x}}_t = (\mathbf{W}^T \mathbf{W} + a \Gamma^T \Gamma)^{-1} \mathbf{W}^T \Delta \mathbf{r}_t \quad (9)$$

V. EXPERIMENTAL VERIFICATION

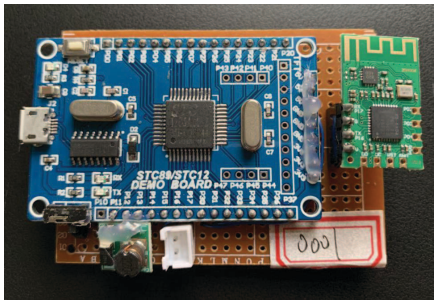
In this section, we introduced the construction of the experimental platform, the experimental scheme and the experimental results, finally compared the traditional methods.

A. Experimental platform

The template is designed so that author affiliations are not repeated each time for multiple authors of the same affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization). This template was designed for two affiliations.



(a) Node principle



(b) Node object

(c)Simulated granary

(d)Experimental vessel

Figure 8..Experimental equipment

B. Experimental scheme

In order to further explore the RF grain system performance, we developed the following three experimental programs in the experimental environment.

1) Experimental scheme 1

On the basis of the feasibility experiment, we continue to explore how different experimental results can be brought by different abnormal grain condition under the condition of 6 sensor nodes. The experimental scheme is

shown in Table 1.

TABLE I. EXPERIMENTAL SCHEME 1

Abnormal	volume	Experimental content
water	500ml	In the off-line training phase of RTI-Grain system, 10 minutes of normal grain RSS data are collected, and then the experimental utensils with abnormal grain condition are successively placed in the center of the granary to collect 10 minutes of RSS data.
air	500ml	
metal	500ml	

2) Experimental scheme 2

On the basis of experiment scheme 1, we continue to explore how different experimental results are brought by the same abnormal grain condition in different locations under the condition of 6 sensor nodes. The experiment scheme is shown in Table 2.

TABLE II. EXPERIMENTAL SCHEME 2

Abnormal	Position	Experimental content
500ml metal	（7.5， 5）	In the off-line training phase of RF -Grain system, the RSS data of 10 minutes of normal grain is collected, and then the glassware with abnormal grain condition is put into five coordinate positions in turn to collect the RSS data of 10 minutes. Among the long axis of granary is x-axis.
	（7.5， 15）	
	（15， 10）	
	（22.5， 5）	
	（22.5， 15）	

3) Experimental scheme 3

On the basis of the above experiments, we continue to explore the detection success rate of three abnormal grain condition in the case of 6 sensor nodes. The experimental scheme is shown in Table 3

TABLE III. EXPERIMENTAL SCHEME 3

Abnormal	volume	Experimental content
water	500ml	In the off-line training phase of RF -Grain system, 10 minutes of normal grain RSS data is collected, and then the abnormal grain condition is put in the center of the granary in order to collect 2 minutes of RSS data. 1000 experiments in each group
air	500ml	
metal	500ml	

C. Experimental results and analysis

The experimental platform is built in the grain Key Laboratory of Henan University of Technology, and the specific experimental scene is shown in Fig. 9.

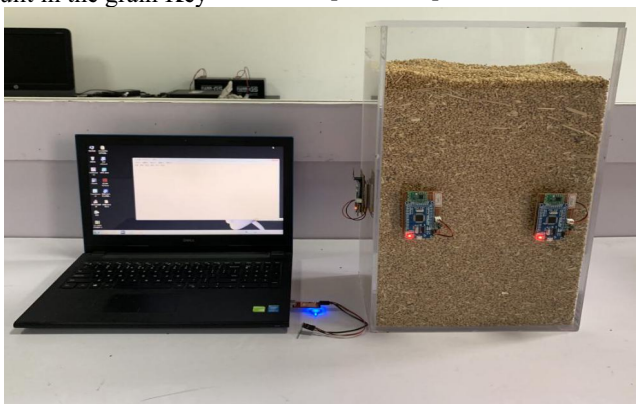
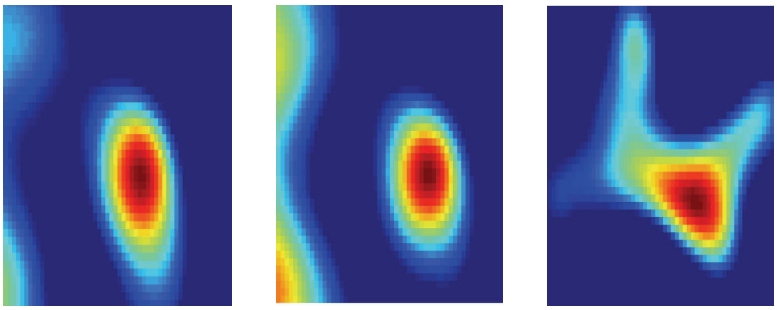


Figure 9. Experimental scene

the results of Experiment are shown as Fig. 10. It can be seen from the Fig. 10, when the abnormal grain condition is metal, the reconstructed image has fewer false positions and image quality. This is because metal has larger dielectric

constant which results in more obvious absorption and attenuation of RF signal



(a)Air reconstruction (b)Water reconstruction (c)Metal reconstruction

Figure10. Experiment 1 Results

The experimental location of abnormal grain condition is shown in Fig. 11, and by the proposed locating method, the mean square error(MSE) of five different positions is shown in Fig. 12. We can see that location MSE of position 3 is the smallest, the main reason is there are more links transmission passing the position 3, and more valuable values for the image reconstruction

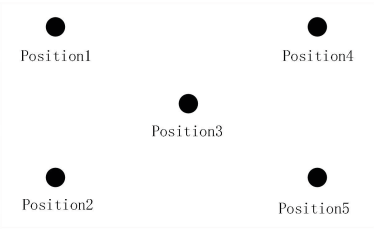


Figure 11. Position hint

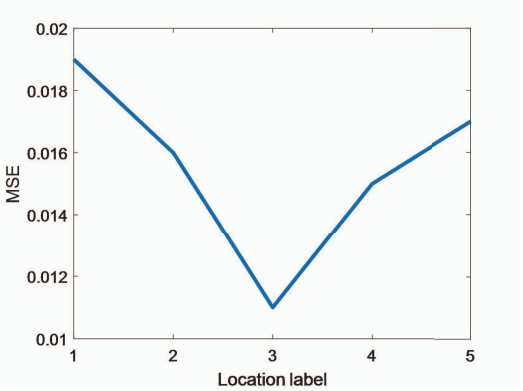


Figure12. location error of different positions

The test success rate of Experiment 3 is shown in Table 4. Metal can absorb more wireless signals and cause more obvious attenuation, thereby which has the higher detection success rate.

TABLE IV. DETECTION SUCCESS RATE

Abnormal	Experimental result	
	Success times	Success rate
water	962	96.2%
air	969	96.9%
metal	981	98.1%

VI. CONCLUSION

In this paper, we propose a non-destructive, low-cost, non-contact method and system based on RSS. The proposed system can detect whether there is abnormal body in Granary and compute find the location of abnormal grain condition by reconstruction imaging. Through the experimental verification, the proposed detection method is completely feasible, and the detection rate is as high as 98%.

ACKNOWLEDGMENT

This research was financially supported by National Science Foundation of China(61871176), Key projects of Henan science and Technology Department (202102110265)； Applied research plan of key scientific research projects in Henan colleges and Universities(19A510011), Scientific Research Foundation Natural Science Project In Henan University of Technology （2018RCJH18).

CONFLICTS OF INTEREST

The authors declare that there is no conflict of interest regarding the publication of this paper.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] Qin Y, Chen J,Fang Y. Non-Dstructive Detection of Barn Reserves Information[J]. Journal of the Chinese Cereals and Oils Association, 2010, 25(4):51-55.

[2] ARMSTRONG Paul. Wireless data transmission of networked sensors in grain storage[C]// The 2003 ASAE Annual International Meeting. July27- 30 2003, Riviera Hotel and Convention Center Las Vegas, Nevada, USA: ASABE2003: 1-12.

[3] Lian, Fei-Yu; Li, Qing. A new recognition method for subsurface targets based on ground penetrating radar map . Source: International Journal of Digital Content Technology and its Applications, 2011,5(8): 31-42.

[4] Wilson J, Patwari N. Radio tomographic imaging with wireless networks[J]. IEEE Transactions on Mobile Computing, 2010, 9(5): 621-632.

[5] Guan S, Song Y, Mu T, et al. Radio Tomography Imaging Based on Distributed Wireless Networks and Experimental Research[J].Journal of Radars,2014,3(4):490-495.

[6] Lee D, Berberidis D, Giannakis G B. Adaptive Bayesian Radio Tomography[J]. IEEE Transactions on Signal Processing, 2019, 67(8):1964-1977.

[7] Zhu C , Wang J , Chen Y . ARTI (Adaptive Radio Tomographic Imaging): One New Adaptive Elliptical Weighting Model Combining with Tracking Estimates[J]. Sensors, 2019, 19(5).

[8] Alippi C, Bocca M, Boracchi G, et al. RTI Goes Wild: Radio Tomographic Imaging for Outdoor People Detection and Localization[J]. IEEE Transactions on Mobile Computing, 2014, 15(10):2585-2598.

[9] Anderson C R, Martin R K, Walker T O, et al. Radio Tomography for Roadside Surveillance[J]. IEEE Journal of Selected Topics in Signal Processing, 2014, 8(1):66-79.

[10] Mostofi Y. Cooperative Wireless-Based Obstacle/Object Mapping and See-Through Capabilities in Robotic Networks[J]. IEEE Transactions on Mobile Computing, 2013, 12(5):817-829.

[11] Bocca M , Kaltiokallio O , Patwari N . Radio Tomographic Imaging for Ambient Assisted Living[C]. International Competition on Evaluating AAL Systems through Competitive Benchamarking,2012:108-130

[12] Bocca M, Kaltiokallio O, Patwari N, et al. Multiple target tracking with RF sensor networks[J] . IEEE Transactions on Mobile Computing, 2014, 13(8): 1787-1800.

[13] Wang J, Gao Q, Wang H, et al. Device-Free Localization With Multidimensional Wireless Link Information[J]. IEEE Transactions on Vehicular Technology, 2015, 64(1):356-366.

[14] Anderson C R, Martin R K, Walker T O, et al. Radio tomography for roadside surveillance[J]. IEEE Journal of Selected Topics in Signal Processing, 2014, 8(1):66-79.

[15] Ojha T , Misra S , Raghuwanshi N S . Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges[J]. Computers and Electronics in Agriculture, 2015, 118:66-84.

[16] Gonzalez-Ruiz A, Ghaffarkhah A, Mostofi Y. An Integrated Framework for Obstacle Mapping With See-Through Capabilities Using Laser and Wireless Channel Measurements[J]. IEEE Sensors Journal, 2013, 14(1):25-38.

[17] Mager B, Patwari N, Bocca M. Fall detection using RF sensor networks[C]// 2013 IEEE 24th International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), 2013: 3472-3476

[18] Morgan D R. On Level-Crossing Excursions of Gaussian Low-Pass Random Processes[M]// On level-crossing excursions of Gaussian low-pass random processes. 2007:3623-3632.

[19] Kondo Y, Numada M, Koshimizu H, et al. Low-pass Filter Without the End Effect for Estimating Transmission Characteristics—Simultaneous attaining of the end effect problem and guarantee of the transmission characteristics[J]. Precision Engineering, 2017, 48:243-253.

[20] Dong Y, Dragut E C, Meng W. Normalization of Duplicate Records from Multiple Sources[J]. IEEE Transactions on Knowledge & Data Engineering, 2018, PP(99):1-1.

[21] Rappaport T S . Wireless Communications: Principles and Practice[M]. Prentice Hall PTR, 2010.

[22] Chiu C Y, Dujovne D. Experimental characterization of radio tomographic imaging using Tikhonov's regularization[C]// Biennial Congress of Argentina. 2014.

[23] Ji Y, Zhu L, Sun X, et al. Differential evolution algorithm with regularization to solve ill-posed equations[J]. Systems Engineering and Electronics, 2018, 40(7):162-166

Design of Laser Marking Control Software based on C#

Linyu Zhu
School of Information Science and Technology
Nantong University
Nantong, China
15706295278@163.com

Yongjie Yang
School of Information Science and Technology
Nantong University
Nantong, China
yang.yj@ntu.edu.cn

Haitao Ye
School of Information Science and Technology
Nantong University
Nantong, China
2418170957@qq.com

Wanting Ren
School of Information Science and Technology
Nantong University
Nantong, China
609561628@qq.com

Xingjia Zhang
School of Information Science and Technology
Nantong University
Nantong, China
1058059601@qq.com

Minghua Sheng
School of Information Science and Technology
Nantong University
Nantong, China
614100717@qq.com

***Abstract*—Laser marking technology has become a popular way of product information marking in industrial production. In order to improve the speed, accuracy, stability and other indicators of the laser marking machine, this paper realizes the design of an intelligent laser marking control software. Users can draw basic graphics in the drawing area. The software adopts the vector marking method. For the drawn content, you can perform basic operations such as movement, rotation and scaling without distortion. After the editing is completed, the software converts the graphics data in the drawing area into the marking data required by the lower computer through interpolation algorithm. After receiving the trigger signal sent by the PLC, the corresponding marking data and operation instructions are transmitted to the control board of the lower computer via USB, and the board drives the galvanometer deflection and laser energy output to complete the marking.**

Keywords-laser marking; control software; vector marking; interpolation algorithm

I. INTRODUCTION

In industrial production, it is often necessary to mark the production date, shift batch number, serial number, etc. on the product to achieve effective production management. Before the advent of laser, inkjet, mechanical or electrochemical methods were generally used for labeling. However, these methods have slow marking speeds, unstable performance, narrow application range, and pollution. Since the birth of the laser in 1958, laser technology has developed rapidly. Laser marking machine is an integrated product of optical, mechanical, and electrical products that comprehensively applies laser technology and computer control technology. It can easily change the marking content. Because of its characteristics such as fast marking speed, fine marking, no abrasion of the workpiece, no pollution, and good durability[1], it stands out among several marking technologies and gradually replaces traditional marking methods. In particular, the galvanometer scanning laser marking machine is widely praised by users for its good optical path sealing, strong adaptability to the environment and high marking quality[2].

The performance indexes of laser marking machine mainly include marking speed, marking accuracy, stability and so on. Since the marking machine does not have strict requirements on the performance of laser equipment, these performance indicators can be improved through the control software[3]. At present, the control methods of the marking machine mainly include single-chip microcomputer control and host computer control[4]. The former promotes miniaturization and reduces costs, but due to the lack of an operating system, there are limitations in software and hardware processing. The latter has strong data processing ability, but it has a certain impact on the stability of the system because the control card is directly installed on the bus slot of the PC. In this paper, the improved control mode of the host computer is adopted. The host computer and the control card are connected via USB, and the host computer completes user content input and data processing. The lower computer directly controls the external equipment to achieve marking. It can effectively avoid the unstable phenomenon caused by the control card installed directly in the bus slot of the PC.

The existing laser marking machines on the market can realize the marking of simple content, but the marking content needs to be manually added, and the start of marking and other actions also require manual operation. The production efficiency is low, which is not conducive to integrated line marking. Therefore, a laser marking control software suitable for industrial production environment is designed. You can draw basic graphics, text and import vector graphics files according to your needs. The software supports the assembly line production function and sets the marking content format. After each marking is completed, the software can automatically switch to the next marking content. The user can choose whether to mark the clear code or the dark code, and only need to input the content, and the corresponding two-dimensional code will be generated automatically by the software. The software directly detects

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00019

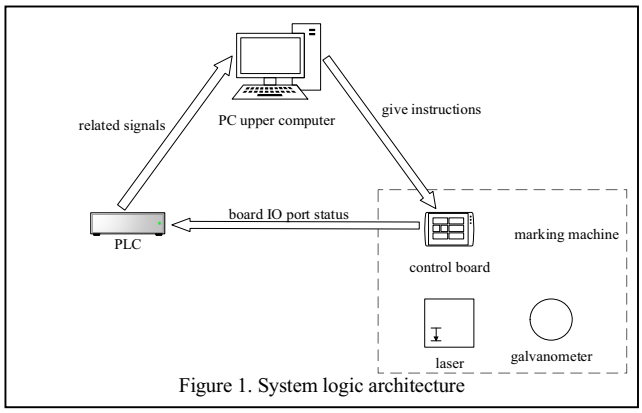


Figure 1. System logic architecture

the trigger signal sent from the PLC and sends the corresponding command to the lower computer to complete the marking operation. After each marking, the information of the marking content will be automatically uploaded to the database. When performing a new marking task, if the user selects the anti-weight function, the software will automatically compare the marking content with the existing content in the database before starting marking. If the content already exists in the database, the user will be prompted to re-enter the marking content.

II. System Logic Architecture

According to the demand analysis of the laser marking machine system, the logical architecture of the system is designed, as shown in Figure 1. The whole system is mainly composed of three parts: programmable control part, host computer control software part and marking machine hardware body. The programmable control part adopts PLC for program control[5], which is used to send related trigger signals to the host computer. The host computer software part mainly completes the design of human-computer interaction interface, drawing and editing of marking graphics, the setting of system parameters, the generation of marking data and transmission of marking data to the control board. The hardware body of the marking machine includes control board, laser, galvanometer, etc. There are generally two galvanometers, which are installed perpendicularly and orthogonally to a motor that can rotate. The control board of the lower computer receives and stores the marking data transmitted from the upper computer. After the data reception is completed, the marking data is transmitted to the laser and the X-Y galvanometer respectively, which are used to drive the laser output laser energy and drive the motor to rotate, and change the deflection angle of the two galvanometer[6], so as to generate laser and make the laser move according to the preset trajectory to complete the marking.

The operation process of the entire system is as follows: After the user completes the drawing of basic graphics and text or imports a vector file from outside, the software converts the graphics data into marking data that can be recognized by the marking machine. The PLC sends the trigger signal such as start or stop to the PC through the agreed communication protocol. After the host computer receives it, the marking data and operation instructions are transmitted to the control board of the lower computer through USB. The control board runs the marking program, drives the galvanometer deflection and laser output energy, so that the marking machine can start marking or stop marking. During the working process of the marking machine, the status of the relevant IO port of the control board is output to the PLC, so as to realize the prompting of the working status of the preparation, working, and completion of marking.

III. Software Design

This article mainly studies the design of the laser marking control software. The software is based on the Window operating system, with Visual Studio as the integrated development environment, and developed using the object-oriented C # programming language. The application program is designed with modularity, and the dependencies between the modules are low, which is convenient for debugging and maintenance[7]. The software has a good man-machine interface, simple and easy to operate, while achieving flexible modification of basic functions such as marking patterns, it can facilitate the expansion of other functions.

A. the Overall Structure of the Host Computer Software

The overall structure of the host computer software is shown in Figure 2. Users can draw basic graphics, characters, Chinese characters, etc. in the drawing area or directly import vector files. The background of the application will store the contents of the drawing area in the form of vector graphics in the graphics data storage area. When the user edits the existing content in the drawing area by moving, reducing, or enlarging, the data in the graphic data storage area will be modified accordingly. After the user revises the drawn graphics, and after the relevant data processing, the graphic data is converted into the marking data required by

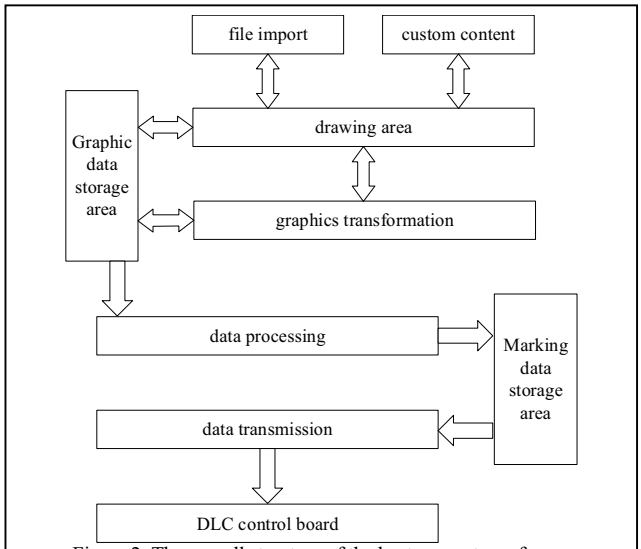


Figure 2. The overall structure of the host computer software

the laser marking machine and stored in the marking data storage area. When the user needs the laser marking machine to mark, the data in the marking data storage area is then transmitted to the DLC control board, and the lower computer drives the laser and galvanometer to print out the graphics according to the data received by the board.

B. Host Computer Software Operation Interface

According to the process of laser marking operation and the analysis of the operating status of the entire marking system, the operation interface of the host computer software of the laser marking machine is laid out, as shown in Figure 3. The menu bar provides function entrances for most functions of the software, such as general file operations, system settings, window display, etc. The toolbar is equipped with operation buttons for the main functions of the host computer, including file management operations such as creating new processing files, importing and saving external vector files, drawing operations for basic graphics and text, editing operations such as the movement, rotation, scaling, mirroring of graphics, as well as undo and restore functions. The drawing area is used to draw and display basic graphics and text added by the user. The object list area describes the name and type of each object added in the drawing area. The object attribute editing area is used to change the position, size and other attributes of the object. The work control area of the board can perform red light mark display or engraving operation. Since the power and frequency of the laser will affect the color and depth of marking, a marking parameter area is set to set the power, frequency, marking speed, etc.

C. Marking Content Drawing

The program uses the GDI+ graphics device interface, which can map the content we draw in the logical coordinate system to the physical device. The core of GDI+ is the Graphics class, which encapsulates some basic graphics and text drawing methods, supports the use of various fonts, font sizes and styles to display text, just install the desired font into the font library of the host operating system itself. Because the basic information of graphics such as lines, ellipses, rectangles, etc. and text is different, the drawing method is different, and the method of saving data is also different. At the same time, considering the readability and maintainability of the program design, the two basic classes of basic information of graphics and graphics tools are defined as BaseShape and BaseTool by using the inheritance of object-oriented language classes. The inheritance of object-oriented language classes is used The information of all graphics is derived from the base class BaseShape and various operations on different graphics are derived from the base class BaseTool, and the respective operations of the graphics are encapsulated by virtual functions. In this way, the relevant data of the graphics are separated from the basic operations, and they are used in combination with each other. While the graphics are drawn and edited, the coupling of the program is reduced to a certain extent, making the program intuitive and clear.

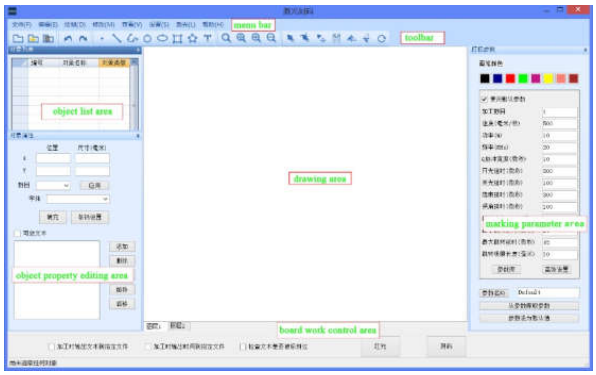


Figure 3. Host computer software operation interface

Taking the drawing of an ellipse as an example, the EllipseTool class inherits the BaseTool base class and rewrites all mouse information about graphics operations in the base class. After clicking the ellipse drawing button on the toolbar, press the left mouse button in the drawing area. Use the mouseDown() function to determine the position of the upper left corner of the ellipse border, then drag the mouse to adjust the size of the border and release the mouse at the appropriate position. Record the coordinates of the point where the mouse is raised by the mouseUp() function, determine the border of the final ellipse, and save the data to the EllipseShape class to complete the drawing of the ellipse. The key code for drawing an ellipse is:

```
Graphics g = this.CreateGraphics();
g.DrawEllipse(new Pen(Pencolor, Penwidth), P1.X, P1.Y, EllipseWidth, EllipseHeight);
```

The first parameter is the Pen object, which is used to determine the color, width and style of the curve. The second and third parameters are the X and Y coordinates of the point pressed by the mouse. The last two parameters are the width and height of the ellipse frame, which are obtained from the absolute values of the X and Y coordinate differences between the mouse down point and the lifting point.

For the drawing of barcodes, take QR Code as an example. Through the EncodeData(char *StringSource) function, the input content is converted into binary, and the result is saved into the global variable two-dimensional array mData[mSize][mSize]. After connecting all the adjacent blocks with the code value "1" in the array and filling them, the desired two-dimensional code can be obtained.

In the Baseshape base class, a hot spot is set for the drawn graphic, which is used to capture the graphic. When you need to edit a figure, you need to select the figure first, so in the Shape class of the figure, it will judge whether the current position of the mouse click is inside the figure, if so, you can change the position of the figure by moving the mouse. Through the hot spot of the figure, we can enlarge, shrink, rotate and so on. After the corresponding editing, rewrite the graphics capture and drawing, so that the graphics can be edited next time.

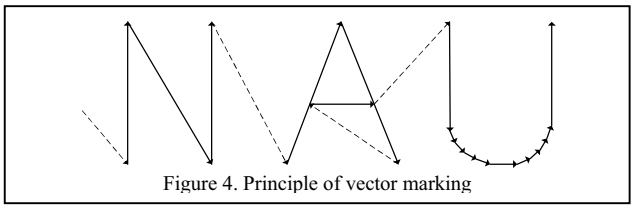


Figure 4. Principle of vector marking

D. Software Algorithm

The software adopts the marking method of vector marking, that is, using directed line segments to represent various graphics and text. The principle of vector marking is shown in Figure 4. The character NAU is described by vector, where the dashed line indicates the pen lifting jump, and the solid line represents the pen falling dash. For the graph composed of curves, the curve needs to be approximated by several line segments[8]. During the marking process, simply reduce the laser power during the jump state or directly turn off the laser so that it does not produce visible traces on the surface of the processing device, and increase the laser intensity during the scribing state to leave marks on the surface of the workpiece, so that the reproduction of the marked content can be achieved. This marking method has the advantages of high efficiency and good graphic accuracy, and there is no distortion or deformation when editing graphics, which greatly improves the speed and quality of laser marking.

The graphic data of the user's drawing area cannot be directly recognized by the marking machine. In order to depict the marking content, it needs to be digitized to form a straight line marking path through which the laser can pass between the adjacent points that need to be marked[9]. Therefore, it is necessary to extract other points that can be output as galvanometers from the original vector through linear interpolation processing.

Linear interpolation approximates the theoretical trajectory by continuously repeating the point-by-point calculation and determining the deviation from the actual, and then performing the direction correction process. The traditional linear interpolation process is shown in Figure 5(a). Taking the first quadrant as an example, for a line segment OP to be drawn, set the coordinates of the end point P to (Xe, Ye), and the coordinates of any interpolation point M For (Xs, Ys), a deviation function can be constructed, that is, formula (1). When F(Xs,Ys) = 0, the interpolation point just falls on the straight line. When F(Xs,Ys) <0, the interpolation point is in the area below the line, and a pulse equivalent length must be corrected in the +Y direction. When F(Xs,Ys)> 0, the interpolation point is in the area above the line , need to be corrected in +X direction. In this way, until the movement reaches the end of the contour. The total number of interpolation points is Xe+Ye. This algorithm has a uniform output pulse, and the operation is relatively intuitive, which is commonly used in laser marking systems. However, there are problems such as slow feed speed and large processing error.

$$F(X_s, Y_s) = Y_s X_e - X_s Y_e \quad (1)$$

In order to reduce the amount of interpolated data and reduce the degree of distortion, this design is optimized on the basis of the traditional linear interpolation method. Still taking the first quadrant as an example, the improved linear interpolation process is shown in Figure 5(b). The slope of the straight line OP is less than 1. For any interpolation point, consider feeding in the +X direction or diagonal direction. Compare the deviation between the two and the actual contour. The one with the smaller deviation is the next feeding direction. The deviation function when feeding in the +X direction is formula (2). The deviation function for feeding in the diagonal direction is formula (3).

$$F(X_{s+1}, Y_{s+1}) = Y_s X_e - (X_s + 1) Y_e = F(X_s, Y_s) - Y_e \quad (2)$$

$$F(X_{s+1}, Y_{s+1}) = (Y_s + 1) X_e - (X_s + 1) Y_e = F(X_s, Y_s) + X_e - Y_e \quad (3)$$

The length of AE in Figure 5(b) is a pulse equivalent, that is, AE = AC + CE = 1. If AC>0.5, CE<0.5. In the right triangle CDE, CE> DE, we can get DE <0.5. This interpolation method makes the interpolation error controlled within 0.5 pulse equivalents, so that the generated contour line is more consistent with the actual contour, and the marking accuracy is effectively improved. And it is not difficult to see that the number of improved interpolation points is less, thus reducing the time for transmitting data to

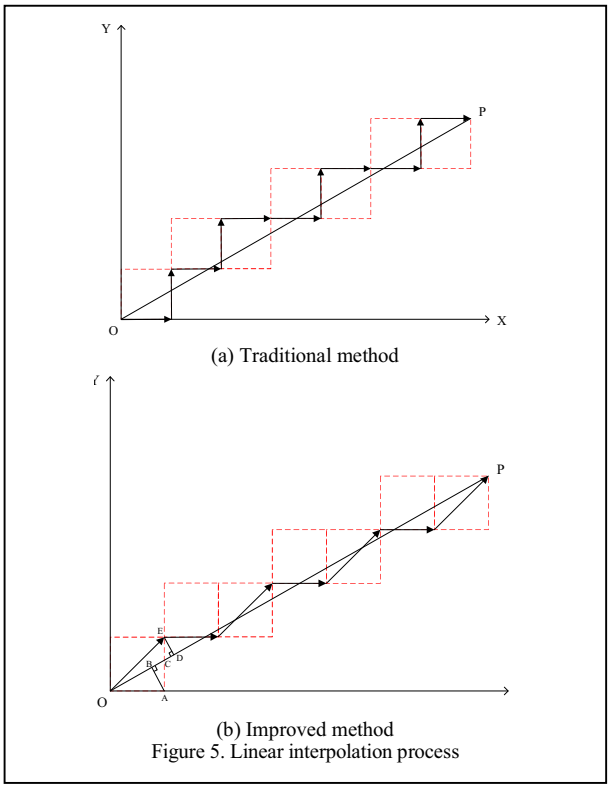


Figure 5. Linear interpolation process

the lower computer. It can better meet the requirements of speed and accuracy in laser marking.

IV. Data Transmission

A. Data Transmission Mode

In order to meet the requirements of the laser marking system for real-time and error-free transmission of marking data, data transmission is performed between the upper computer and the lower computer via USB. Based on USB's hot-swap feature, the computer can directly plug and unplug the USB connected to the DLC control board without shutting down the computer. After the board is connected to the computer, the software automatically detects and completes the relevant configuration. Without manual intervention, it is plug-and-play, enabling multiple marking devices to be connected to the PC at the same time.

The serial data in the USB bus is encoded in the form of NRZI, and the data is transmitted as a differential signal[10], which can largely resist the electromagnetic interference generated during the laser marking process. In addition, it adopts the batch transmission format of the USB transmission type, which can transmit large blocks of data, process and correct transmission errors in the form of error detection and retry to ensure the accuracy of the data.

B. Data Transmission Format

During the laser marking process, the DLC control board needs to perform D/A conversion on the received marking data sent from the host computer to obtain an analog voltage signal that controls the deflection of the servo motors in the X and Y directions, thereby driving the galvanometer deflection. At the same time, the laser output power is controlled during the deflection process, so as to achieve the desired marking effect. In the laser marking process, the data that the host computer needs to transmit are mainly divided into two types. The first type is the control command data, which mainly includes the start marking signal, the stop marking signal, and the working mode signal of the marking machine. Such data is generally small and the transmission format is relatively simple. The second type is the interpolation data of the drawn vector graphics, including the X and Y axis positioning signals of the galvanometer, the laser power signal, the acousto-optic Q switch control signal, the data output delay control signal between the marking points, etc. Because the D/A conversion chip on the control board is 16 bits, the X and Y axis positioning signals of the galvanometer are each represented by two bytes of data. The power control signal of the laser is also represented by two bytes. For the acousto-optic Q switch signal, it is represented by one byte, but only one bit is actually used, so the lowest bit can be set to be effective. The delay signal is determined by the interpolation cycle and is also represented by one byte. The data transmission format of each interpolation point is shown in Figure 6. The marking data transmission sequence is based on the data of the first interpolation point, the data of the second interpolation point ... the data of the last interpolation point.

XL	XH	YL	YH	PL	PH	Q	T

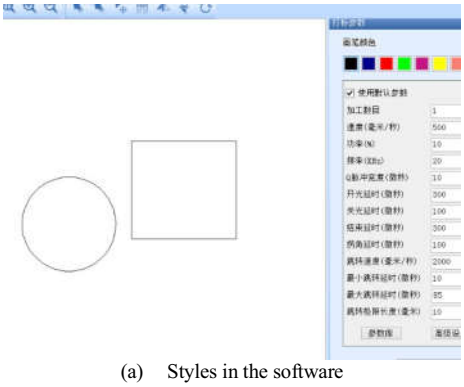
注：XL：lower 8 bits of galvanometer X-axis positioning control signal
XH：upper 8 bits of galvanometer X-axis positioning control signal
YL：lower 8 bits of galvanometer Y-axis positioning control signal
YH：upper 8 bits of galvanometer Y-axis positioning control signal
PL：lower 8 bits of laser power control signal
PH：upper 8 bits of laser power control signal
Q：acousto-optic Q switch control signal
T：delay control signal

Figure 6. Data transmission format for each interpolation point

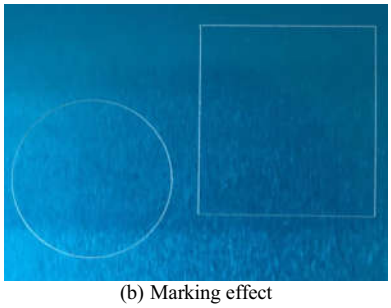
V. Software Test

The test method for basic graphics is the same, so here we take circle and rectangle as examples. Firstly, draw a circle and a rectangle in the drawing area of the software, edit the position and size of the graphics, and set the relevant marking parameters, as shown in Figure 7 (a). Then click the red light button to adjust the position of the area where the marking content is displayed on the test card. After confirming, turn off the red light and click the marking button to mark. The marking effect is shown in Figure 7 (b). It can be seen that the circle and rectangle marked are consistent with those displayed in the drawing area of the software.

Next is the marking test of the two-dimensional code and text. Select the QR Code font in the barcode font, and enter the content of "Hello 2020" in the text box, choose to display both the clear code and the dark code at the same time, click the fill button to fill, and the parameters settings



(a) Styles in the software



(b) Marking effect

Figure 7. Graphic marking test

(a) Styles in the software

(b) Marking effect

Figure 8. Two-dimensional code and text marking test

still use the default parameters, as shown in Figure 8 (a). The remaining operation steps are the same as the graphic marking test. The final marking effect is shown in Figure 8 (b). The printed content is the same as the content added in the software, there is no blur and distortion, and the expected marking effect can be achieved.

VI. Conclusion

This design has studied some key technologies in the laser marking control software, and applied the designed software to the laser marking system. The system runs stably and the marking accuracy is high. Through this software, users can draw basic graphics, texts, generate two-dimension codes or import vector files directly from the outside, and convert them into data files required by the marking machine. The human-machine interface is friendly. For the input text, there is a selection function of the clear code and the secret code, and the user can select the format of the marked text content as needed. At the same time, the software of the host computer improves the trigger signal control, supports the automatic generation and switching of the marking content according to the set format, and improves the efficiency of the coding machine. In addition, the software introduces the application of the database, with anti-duplication function, which can meet the unique requirements of the marking information in the production process of the product and is suitable for industrial production sites.

Acknowledgment

I would like to extend my deep gratitude to all those who have offered me practical, cordial and selfless support in writing this thesis. Especially my tutor, Professor Yang, who influenced me with his insightful ideas and meaningful inspirations, guided me with practical academic advice and feasible instructions. Besides, the authors thank the three anonymous reviewers for their helpful suggestions.

References

[1] Z. Guoping, L. Pingwei, C. Shengyang and Z. Bo, "Design of laser marking controller based on ARM+FPGA," 2010 International Conference on Electrical and Control Engineering, Wuhan, pp. 752-754, 2010.

[2] W. Lihe and M. Dianguang, "Development trend of laser marking machine and design of its control system", Electromechanical Information, vol. 8, pp. 103-104, 2015.

[3] O. Xiaoliang, "Research and development of laser cutting machine control system software", China Equipment Engineering, vol. 1, pp. 219-220, 2019.

[4] T. Lin and W. Yinan, "Communication design of host PC and PLC", Science and Technology Innovation, vol. 34, pp.33, 2016.

[5] N. Eichner, "Laser marking of small manufacturing lots at high throughput", Laser Technik Journal, vol. 2, pp. 36-37, 2017.

[6] S. Xiao, "Partition genetic algorithm in laser marking of graph", Guangdianzi Jiguang/Journal of Optoelectronics Laser, vol. 2, pp.223-229, 2018.

[7] V. Cherkasova and O. Brikova, "Development and modeling of an automatic drive micropositioning control system for laser equipment positioning tasks," 2019 III International Conference on Control in Technical Systems (CTS), Russia, pp. 118-120, 2019.

[8] P. Junguo, X. Xinping and W. Shengqiang, "A universal interpolation algorithm for parametric curves," 2010 International Conference on Electrical and Control Engineering, Wuhan, pp. 311-314, 2010.

[9] H. Nie and H. Chen, "Piecewise linear interpolation algorithm in the high precision electronic system," 2018 5th International Conference on Systems and Informatics (ICSAI), Nanjing, pp. 65-69, 2018.

[10] Y. Pai, F. Cheng, S. Lu and S. Ruan, "Sub-trees modification of huffman coding for stuffing bits reduction and efficient NRZI data transmission," in IEEE Transactions on Broadcasting, vol. 58, no. 2, pp. 221-227, June 2012.

Network Entity Landmark Mining Technology

Yong Gan
Zhengzhou Institute of Engineering and Technology
Zhengzhou, China
e-mail: ganyong@zzuli.edu.cn

Yuanbo Liu
School of Computer and Communication Engineering
Zhengzhou University of Light Industry
Zhengzhou, China
e-mail: zzulilyb@163.com

Helin Zhang
School of Computer and Communication Engineering
Zhengzhou University of Light Industry Zhengzhou, China
e-mail: zhanghelin5460@foxmail.com

Dongwei Jia
School of Computer and Communication Engineering
Zhengzhou University of Light Industry
Zhengzhou, China
e-mail: 602914623@qq.com

***Abstract*—With the increasingly prominent role of network entity landmarks in IP geolocation and cyberspace security, research on mining technology of network entity landmark is becoming more and more in-depth. Based on this, we will focus on the application and methods of network entity landmark mining (both traditional and more recent). This paper first outlines the basic concepts and applications of network entity landmark mining technology; then, we divide the existing landmark mining algorithms into two types: city level and street level, and analyze each type of typical algorithm; finally, we made a comprehensive comparison of some network entity landmark mining algorithms.**

Keywords-component; landmark mining; landmark evaluation; Web; geographic location

I. INTRODUCTION

Today, with the rapid development of information, the issue of privacy leakage on the Internet is emerging one after another, and the security of cyberspace has become particularly important. Network entity positioning technology is one of the important technologies in cyberspace security, referred to as IP geolocation, which refers to determining the geographic location of IP devices in the Internet network[1,2]. This technology plays a key role in network security applications such as intrusion detection[3], network attack tracing[4], and network diagnosis[5]. It also has a wide range of applications in many civil fields, such as weather forecasting, web page language Automatic matching, targeted advertising placement, regional-based access control, network management, and network performance optimization[6], etc.

Many research institutions and scholars have carried out in-depth research in the field of IP geolocation, and put forward a series of IP geolocation methods[6,7,8,9,10,11,12], in which landmark-based IP positioning technology has been widely recognized for its high positioning accuracy and positioning reliability. This technology uses the network connection relationship between the landmark and the positioning target and the geographic location of the landmark to infer the target's location. The current localization methods based on landmarks include SLG[13], NNBG[14] and a series of improved methods. The accuracy of this method largely depends on the density and accuracy of network entity landmarks.

With the application of IP geolocation in the field of Internet of Things, to improve the positioning accuracy and precision, there are new requirements for the number, coverage and accuracy of network physical landmarks. However, due to the complexity of the Internet, the closure of commercial data, and privacy protection, high-precision landmark mining methods still have problems with insufficient number of landmarks, limited coverage, and low accuracy. The landmarks excavated by different means can complement and confirm each other to expand the coverage of the landmarks, but there are a lot of inconsistent locations of the landmarks. Most current positioning methods rely on single-source landmarks, failing to comprehensively utilize multi-source landmarks to improve positioning range and accuracy. Therefore, the current high-precision landmark mining cannot meet the needs of high-precision IP positioning. It is of great practical significance to continue the research of network entity landmark mining technology.

II. RELATED WORK

The research of landmark mining technology mainly focuses on collecting landmarks for public databases, log files, Internet resource services and terminal data sources. According to the location accuracy of acquiring landmarks, this article will introduce landmark mining with different accuracy from two aspects. The research of landmark evaluation technology mainly focuses on two directions: the evaluation of city-level landmarks is mainly aimed at the IP location database landmarks, and the street-level landmark evaluation is primarily aimed at the landmarks obtained by

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00020

the street-level landmark acquisition algorithm. The process of landmark mining technology is shown in Fig.1.

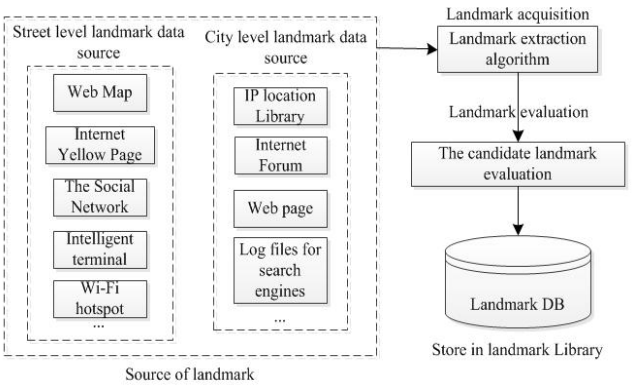


Figure 1. Frame diagram of landmark mining technology.

A. Basic Concept

This section introduces several important concepts involved in the research of network entity landmark mining technology.

Network entity landmark: Referred to as a landmark, it refers to a network device whose geographic location is known and has a clear IP identity that can respond to the detection of a probe point to correct the location of other IPs, usually collected from open source landmark sets such as PlanetLab. According to its geographical location accuracy, network entity landmarks can be divided into city-level landmarks and street-level landmarks.

City-level landmarks: a network entity landmark whose geographic location is accurate to the city. Its geographic location is usually expressed as the name of the city where the landmark is located.

Street-level landmarks: a network entity landmark whose geographic location is accurate to the block. Its geographic location is usually expressed as the longitude and latitude value of the landmark location.

Candidate landmarks: refers to the directly obtained IP address and its associated geographic location information, whose location accuracy needs further evaluation.

Landmark mining: Landmark mining is the process of collecting IP addresses and their geographic location information on the network, and filtering through certain means to obtain accurate network entity landmarks. Landmark mining can be divided into two stages: landmark acquisition and landmark evaluation.

Landmark acquisition: The process of collecting IP addresses and geographic location information on the network, standardizing its format and constructing candidate landmarks.

Landmark evaluation: Landmark evaluation, also called credibility evaluation of candidate landmarks, refers to the process of judging and estimating the accuracy of candidate landmarks by certain technical means, and selecting the network entity landmarks with accurate locations from them.

B. The Structure of Paper

The network entity landmark mining technology has been openly discussed for more than ten years. Researchers have proposed various methods, but due to the different experimental environments (network hosts used for positioning, network landmarks used, etc.), they are different. The research of landmark mining technology is difficult to comprehensively review, and it also lacks theoretical analysis. On the other hand, the review work on landmark mining technology at home and abroad is not perfect. It is difficult for people to establish a comprehensive understanding of the future direction of network entity landmark mining technology.

Section 1 of this article provides a brief overview of the progress of network entity landmark mining technology in network security and its impact on IP positioning. Section 2 outlines the basic concepts of network entity landmark mining technology respectively. Sections 3 and 4 analyze the typical city-level and street-level network entity landmark mining and evaluation algorithms in detail. Section 5 comprehensively evaluates various network entity landmark mining algorithms and challenges. Section 6 summarizes the full text.

III. CITY-LEVEL LANDMARK MINING AND EVALUATION

A. City-level Landmark Acquisition

The current main method of acquiring city-level landmarks is the city-level landmark acquisition based on the IP location database. This method selects multiple IP address segments with the same geographic location in the IP location database and selects candidate landmarks from them. Some companies engaged in IP positioning have constructed and maintained relatively accurate IP location databases. Among them, the more common IP location databases in China include: Baidu, Taobao, IPIP, Evan Technology, and well-known foreign countries include IP2Location, Maxmind, DBIP, etc. The WHOIS database also contains the location information of the IP address block and its owner. These databases have a large number of IP segment location information, the data volume reaches millions, and some city-level location libraries can cover the entire IP space. Such methods have high acquisition efficiency, abundant landmarks and wide coverage. However, this method has inherent drawbacks. On the one hand, the IP addresses in the IP location database are presented as IP address segments, and there are a large number of non-surviving IPs, which is difficult to meet the requirements of the positioning method for the success rate of landmark detection; on the other hand, it is not effective to ensure the accuracy of its data.

Guo C et al. proposed the Structon method based on Web pages[15]. They believed that Web pages were embedded with rich geographic location information, such as administrative area information (province, city, district), zip code, telephone area code, etc. This information is associated with the IP address of the server corresponding to

the Web page to realize the mapping between the geographic location and the IP address, thereby constructing candidate landmarks. This method first processes the web page according to the HTML tag, splits the web page into multiple HTML blocks, and uses regular expressions to extract the geographic location information; then the location information extracted from each HTML block corresponds to the IP of the web server domain name Address association constructs a position weight vector and votes to obtain seed landmarks; finally, heuristic guessing algorithm is used to expand the geographic location information of the seed landmarks in the /24IP segment, and the landmarks are corrected according to AS and BGP information, and voted to determine its location to improve the accuracy and coverage of landmarks. This method is the first to extract location information from Web pages to realize the mining of Web server landmarks. The number of landmarks obtained is large, widely distributed, and good stability. However, there are many difficulties in the implementation of this method. Obtaining massive web resources not only consumes traffic and storage resources but also restricts the speed of landmark mining. On the other hand, different web pages have different languages, different formats, and it is difficult to extract geographic location patterns.

Zhu G et al. proposed a city-level landmark acquisition method based on Internet forums[16]. This method infers the city where forum users are based on the semantic information in the forum name, collects user IP addresses from them, and then builds the association between the two landmarks. The main steps are as follows: First, select a topic forum with strong regionality and relatively concentrated forum users (such as “Zhengzhou University” or “Kaifeng Real Estate” and other forums); then extract all the IP from it, and use multi-bank voting to determine these The city where the IP address is located, and retain the IP with consistent city information; finally, associate the IP address with the city to construct a city-level candidate landmark. This method has a large number of landmarks and a broad coverage. However, it can only mine one city at a time, and the semantics of the forum need to be manually analyzed, so the mining efficiency is not high enough. Especially with the increasing awareness of personal privacy protection, most forums no longer display user IP address information, and the application of this method is greatly restricted.

Hyunsu Mun et al. proposed a landmark acquisition method based on the online trading market[17]. This method collects public seller information from the online trading market platform, obtains its IP address and shipping location information to construct candidate landmarks, and then selects the same landmark The location of the IP address in the /26IP segment is voted to determine its city-level location. Using this method, the city coverage of the Korean landmark database constructed with 29 months of transaction information in the Korean trading platform Ruliweb has reached 90.11%, and the accuracy of landmarks has also been effectively guaranteed. However, most online trading markets hide the seller's IP address and do not disclose fine-grained location information. The data source of this method is limited, and the number of landmarks cannot be effectively guaranteed.

Dan O et al. proposed a landmark acquisition method based on search engine log files[18]. This method extracts the user’s IP address and search content from the search engine log files, and then determines the user’s city based on the search content that contains location information. Finally, aggregate the location information of the city and associate it with the IP address, and correct its location by querying the IP location database to construct a city-level landmark. The verification of the IP data set with known locations shows that the location accuracy of this method is significantly higher than the current latest method, but this method requires a large number of search engine log files, and the singleness of the data source limits the application of this method.

In summary, the city-level landmark acquisition method collects landmarks from commercial IP location libraries, Internet resources, and other sources. The number of candidate landmarks acquired is large and covers a wide range, but the accuracy of landmark locations needs to be further evaluated.

B. City-level Landmark Evaluation

1) City-level Landmark Evaluation Based on Route Identification

Wang T et al. proposed the GeoCop method for city-level landmark evaluation based on edge route identification[19]. This method first deploys a large number of detection sources outside the target area, and performs path detection to collect the correlation between the detection source, the intermediate router, and the target IP. Then, the edge routers close to the target are identified according to the statistical rules of the corresponding relationship between the detection source and the detection target. Finally, the location of the router is determined according to the city-level location of the candidate landmarks associated with the edge router, and then the evaluation and correction of the city-level landmark location can be achieved.

Ma T et al. proposed a city-level landmark evaluation algorithm based on route identification[20]. This algorithm first constructs a routing geographic location dictionary based on the city’s administrative divisions and landmark locations (such as airports) in the target area, and establishes routing host name matching rules. Use multiple detection sources to perform Traceroute detection on candidate landmarks to obtain their path information, and extract the host name of the route on the detection path in combination with the routing host name matching rule. Then, determine the city where the route is located according to the route location dictionary. Finally based on the nearest router and

candidate the relative delay between landmarks determines the reliability of landmark city locations.

Zhu G et al. proposed a city-level landmark evaluation method based on routing hops[21]. This algorithm first queries multiple IP location libraries to determine the cities where candidate landmarks may be located; and then uses multiple pre-configured detection sources, respectively detect candidate landmarks and reference nodes in cities where they may be. Then, based on the number of hops from the detection source in the reference node detection path to the city and the number of hops within the city, establish a reference hop vector for these cities, and create a candidate landmark hop number vector according to the detection path of the candidate landmark; finally, the similarity between the reference hop vector and the candidate landmark hop vector is calculated, and the city corresponding to the reference hop vector with the highest similarity to the candidate landmark is used as the city where the candidate landmark is located.

Such methods have the characteristics of high evaluation efficiency and wide application range, but they need to deploy a large number of detection sources to detect the target host or reference point. In addition, during the evaluation process, algorithms that rely on statistical laws such as voting, speculation, and similarity calculation will limit the accuracy of landmark evaluation.

2) City-level Landmark Evaluation Based on POP Network Analysis

Shavitt Y et al. proposed a city-level landmark evaluation method based on POP network analysis[22]. They first probed the IP address of the target area to obtain network topology data, then analyzed the network topology to find the POP network, and finally used many An IP location database queries the declared position of each IP address of the POP network. If 50% of the declared positions of the IP addresses are within a certain range (100km), then the center of these declared positions is taken as the position of the POP network, and evaluate the city-level location of all IP addresses in the POP network.

Zu S et al.[23] improved the evaluation method of city-level landmarks based on POP network analysis. Before they analyze the POP network, they first divide the network nodes belonging to the target city from the detection path according to the distribution rule of the single-hop delay between the network nodes of different cities, and expand the landmarks; finally, they use the common anonymous routing structure to search and merge the anonymous routes in the path information, to realize the discovery of the POP network where the city-level landmarks are located and the accuracy assessment of the landmark location.

This kind of method starts from the topology of the network, finds the network nodes gathered in a certain area, and evaluates the accuracy according to the declared position of the nodes. The evaluation results are reliable and efficient, but it is only suitable for candidate landmarks evaluation with the POP network structure. The network structure is not universal in the network topology and has certain limitations.

3) City-level Landmark Evaluation Based on Voting

LI H et al. proposed a voting-based city landmark evaluation method[24]. They used multiple IP location libraries to query the location information of candidate landmarks, and then used a voting method to determine the IP location at different granularities, finally performed location verification to achieve city-level location assessment of landmarks. The method has high evaluation efficiency and a large number of landmarks. However, different databases have different degrees of reliability, and the reliability of landmarks in different regions of the same database is also different. In addition, there are copy phenomena between databases, which will lead to poor evaluation of the method.

In summary, the city-level landmark evaluation method combines host name, network topology, multi-database query results and other information to calibrate the city-level location of the candidate landmark, which greatly improves the accuracy of the landmark. However, these methods require a lot of data to support, and there are certain limitations.

IV. Street-level Landmark Mining and Evaluation

A. Street-level Landmark Acquisition

Web landmarks have the advantages of a large number of landmarks, good stability, and open ports. They are an important part of street-level landmarks. In the process of acquiring landmarks, Web landmark information can be obtained from Web maps, Web pages, Internet yellow pages, etc. It does not rely on third-party data, and can achieve automatic acquisition of landmarks under non-collaborative conditions. Therefore, Web landmarks are the main streets currently acquired landmark.

1) Street-level Landmark Acquisition Based on Web Resources

Wang Y et al. proposed a landmark acquisition algorithm based on online maps[13] (Comprehensive Landmark Mining Algorithm, CLMA). They believe that many schools, hospitals, enterprises, government departments and other units have their own Web servers, and often deploy the server inside the unit. Therefore, the location of the organization and the domain name of its Web server can be obtained through online map query to realize the construction of street-level landmarks like Web servers. This method first enters keywords such as “enterprise,” “school,” “hospital” or “government” and the name of the administrative region on a website that provides online map services (such as Baidu maps and Google maps). The map server will return a series of related organization’s website domain name and its latitude and longitude information. Finally, the DNS service is used to convert the domain name of the website into an IP address, and then the mapping relationship between the IP address of these websites and their geographic location is obtained, and candidate landmarks are constructed. The algorithm can get web server landmarks according to the specified city, which has wide coverage and stable performance. However, the algorithm uses the text search service of online map, and there is a maximum number of returned landmarks in each

search, and the results obtained by repeated search are the same, because this method can only get some candidate landmarks in an online map, which is less, so it is difficult to meet the IP geolocation needs.

Yang W et al. proposed a street-level landmark mining algorithm based on radar search. Firstly, the algorithm divides the region to be mined into square sub-regions of appropriate size; secondly, it dynamically adjusts the search granularity and uses the online map radar search service to recursively search each sub-area to obtain all landmark information; finally, street-level candidate landmarks are constructed to fully mine the online map. The number, distribution and location effect of landmarks obtained by this algorithm are significantly improved compared with the existing algorithm based on an online map.

Liu H et al. proposed a checkin-geo method based on the social network check-in data[25], which uses the location data that users actively share in the social network and the user log that they log in from the PC to associate the user's geographic location and IP address to achieve the acquisition of street level landmarks. Compared with the existing methods, the location accuracy of the landmark obtained by this method is one order of magnitude higher, and the landmark is widely distributed with high location accuracy. However, due to the need for privacy protection, the application of this method is limited.

Some researchers proposed a street-level landmark mining algorithm based on Internet Yellow Pages. Considering that the yellow pages contain a large number of web and email domain names corresponding to organizations, this method uses regular expressions to extract the geographical location of organizations and web and email domain names from the yellow pages, and resolves the domain names to corresponding IP addresses. Finally, the IP address is associated with the geographic location to build the candidate landmark. This method is fast for the designated yellow page mining, and the performance of the acquired landmark is stable. Still, the selection of the yellow page and the analysis of the yellow page structure need manual intervention, so the efficiency of landmark mining is not high enough.

2) Street-level Landmark Acquisition Based on Web Resources

Zhao F et al. put forward a landmark acquisition method based on Wi-Fi hotspot[26], they use intelligent mobile devices with GPS positioning function to access Wi-Fi hotspot in the area to be mined, then use preset network nodes to detect Wi-Fi hotspot to obtain the public IP address of its access router, and determine the location of Wi-Fi hotspot according to the location data obtained from GPS positioning. Finally, the public IP address of its access route is associated with the geographical location of Wi-Fi hotspot to form a landmark. Wi-Fi router is widely used, with high density in the city, and the location accuracy obtained by GPS positioning is high, so that this method can ensure the number of acquired landmarks in a given area and the location accuracy of landmarks. However, using intelligent mobile devices to collect Wi-Fi hotspot requires human intervention, so this method is only suitable for a small range of given area, not for a large range of landmark acquisition.

3) Street-level Landmark Acquisition Based on Service Open Port

Li R et al. proposed a street-level landmark mining method based on the service open port[27], which first classifies the IP address according to the service level; then uses the open port of the IP address as the feature to train SVM classifier with manually labeled data, and selects the IP address of the server in a given region; then uses multiple different DNS servers to reverse check the domain name corresponding to the IP address; and After that, they use domain name database, WHOIS database or online map query method to realize the association of domain name and location, and get street level landmark. Compared with the latest method, the time cost of this method is less, and the positioning accuracy is also improved. However, the server IP address only accounts for a small part of the IP space. This method has no prior knowledge of the target area IP, and the scanning efficiency of the IP segment is not high.

4) Street-level Landmark Acquisition Based on Intelligent Terminal

Triukose S and others proposed a landmark acquisition method based on Intelligent Terminal[28], which is from the third-party mobile app Obtain the IP address of the smart phone user and the location information provided by the GPS device, and get the corresponding information of IP address, geographic location and time stamp. Then analyze and calculate the location distribution of the IP address and the street-level location of the IP address according to the time change. With the popularity of intelligent terminals, the mobile terminal type landmarks obtained by this method have greater advantages in quantity and distribution. Still, this method relies on a large number of collaborative data. With the increasing concern of user privacy protection, the application of this method is greatly limited.

To sum up, the location accuracy of the landmark obtained by the street level landmark acquisition method is high, and the location accuracy has certain guarantee. However, these methods are limited by collaborative data sources, acquisition efficiency and other aspects, and the number of landmarks obtained is limited. Therefore, the current street level landmarks obtained by these methods are challenging to meet the needs of street-level positioning.

B. Street-level Landmark Evaluation

1) Street-level Landmark Evaluation Based on Network Measurement

Wang Y and others proposed a street-level landmark evaluation method (LVM) based on Web server landmark verification. This method first verifies the postal area. If the postal code contained in the candidate landmark information is inconsistent with the postal code of its geographical location, the candidate landmark is deleted. Then, it verifies the web page request and accesses the web page through the IP address and domain name provided by the candidate landmark. If the label content obtained is different, the web site is considered to be hosted on the shared host or in the

ECS, delete the candidate landmark; finally, exclude the branches, and remove the candidate landmark with the same domain name but different zip code or geographical location. This method excludes some invalid landmarks such as hosting, sharing host and CDN network, and greatly improves the accuracy of landmarks. However, this method mainly depends on whether the IP and domain names of landmarks can open the same web page to determine whether the location of landmarks is reliable, which may lead to the accurate location of shared host class, cloud service class landmarks and landmarks that do not support IP access to web pages may be deleted by mistake. In contrast, the inaccurate location of CDN network class and managed host class landmarks may be wrongly evaluated as reliable landmarks. These errors limit the accuracy of landmark location to some extent.

Li R et al. [29] proposed a street-level landmark evaluation method based on the nearest common router, which is called SLE for short. This method regards the mechanism as an area, and uses the landmark access router to group it and convert the delay value into distance constraint. The relationship model of geographic distance, distance constraint and regional radius between any two landmarks in the group is established. Finally, the reliability of landmarks is calculated by the binomial distribution. This method can realize the evaluation of street-level landmarks from various sources, but there are two problems in this method. One is the high-density requirement for the candidate landmarks, and there are at least two candidate landmarks to be evaluated under the last hop common route; the other is the low efficiency of evaluation, using the strategy of "taking the minimum from multiple measurements" to calculate the delay value, when the number of landmarks is large, the time cost of multiple network measurements on landmarks is high, which affects the efficiency of evaluation.

Yang W et al. proposed a street-level landmark evaluation algorithm that can estimate the upper limit of landmark error[30]. The algorithm first verifies the location of the city level, then detects the landmark to obtain its last hop route, and groups the candidate landmarks according to its last route. Then, the E-apriori algorithm is used to cluster the declared locations of landmarks in each group. According to the clustering results of each group of candidate landmarks, the location range of the last route and the probability of the route falling in this area are determined. Finally, reliable landmarks are selected according to the location range of the last hop route, and the error range and accuracy are determined.

2) Street-level Landmark Evaluation Based on IP Geolocation

The main idea of the evaluation method of street-level landmark based on IP Geolocation is to locate the IP address of the candidate landmark using the street level IP location technology, then compare the location results with the declared location of the candidate landmark, and select the reliable landmark according to its distance error. According to the positioning results of the candidate landmarks and the errors of the positioning methods, the real location range of the candidate landmarks can be determined (the circular area with the positioning results as the center and the positioning errors as the radius). Then, the maximum error (sum of positioning error and error distance) of the selected candidate landmarks can be calculated by combining the error distance between the declared position of landmarks and the positioning results. The error of landmark obtained by this kind of evaluation method is within a certain range, and the accuracy is high, but it is only suitable for the area with positioning ability.

3) Web Landmark Evaluation Based on Multi-level Decision

Yang W et al. put forward the Evaluator algorithm of Web landmark reliability evaluation. The algorithm adopts the idea of a decision tree to filter the invalid landmarks layer by layer, and comprehensively uses the public data and services to evaluate the candidate landmarks to obtain reliable landmarks with quantitative credibility. Besides, an effective algorithm (DNS distributed resolution algorithm) is proposed to resolve all IP addresses of domain names, which provides data support for the Evaluator to filter candidate landmarks. The reverse verification algorithm is used to obtain all domain names hosted by IP, and the gradient descent method is used to estimate the evaluation parameters, which effectively improves the evaluation effect of the Evaluator. The experimental results show that this method can effectively eliminate the invalid web landmarks, and significantly improve the accuracy and coverage of the standards compared with the current related evaluation methods.

To sum up, the evaluation methods of street-level landmarks mainly use network measurement and street-level IP positioning to evaluate the candidate landmarks. There are still two main problems in these evaluation methods. First, the evaluation effect of important candidate landmarks is not good. Web server landmark is an important Street landmark, but the LVM method, which is specially used for the evaluation of web server landmark, is easy to be deleted and commented by mistake. Other algorithms are not targeted and ineffective due to their inherent limitations. Second, the key performance of landmarks cannot be quantified. The error range is the key attribute that affects the positioning effect. The current methods can not quantify the error range of landmarks and affect the role of landmarks.

V. COMPREHENSIVE ANALYSIS

In the first half of this paper, we have described various typical mining technologies of network entity landmarks. In this section, we will analyze the performance of the algorithm in combination with the typical evaluation

TABLE I. COMPREHENSIVE COMPARISON OF VARIOUS LANDMARK MINING AND EVALUATION TECHNOLOGIES

Algorithm	Positioning accuracy	Accuracy rate	Stability	Source	Restrict
MaxMind	<50km	poor	medium	IP location Library	Low success rate of landmark detection
Structon	<50km	medium	good	Web page	Difficulty in extracting different language formats
FBL	<50km	good	medium	Internet Forum	Most forums do not display user IP
OUMA	<50km	medium	medium	Online trading platform	No guarantee for the number of landmarks
OQL	---	medium	medium	Log files for search engines	Single data source
CLMA	5km<d<50km	good	good	Webserver	Landmarks for mining web server types only
LVM	5km<d<50km	good	good	Webserver	It may be deleted or evaluated by mistake
RLMA	<50km	good	medium	Online map	Single data source
SLE	50km<d<100km	medium	medium	Web resource	long processing times
Evaluator	<50km	medium	good	Web landmark	Small scope of application

indicators. Table 1 shows the comprehensive comparison of some landmark mining and evaluation technologies.

Positioning accuracy: the most important index to measure the advantages and disadvantages of the positioning algorithms. The positioning accuracy of different granularity determines the application scope of the algorithm.

Stability: it is mainly measured from the cost of algorithm design and deployment, the number of landmarks obtained and whether it is easy to deploy incrementally. It is divided into three levels: poor, medium and good.

Algorithm limitations: the shortcomings of the algorithm under certain conditions.

As can be seen from the above table, the development trend of network entity landmark mining technology is as follows:

At present, the main source of street-level landmarks is a web server, which is limited in number and mainly deployed in large and medium-sized cities with developed Internet. Due to the low density of landmarks, small towns and vast rural areas can not meet the positioning needs. Especially in recent years, more and more organizations are hosting the network stations on the cloud, the number of web server landmarks is decreasing, and the source of Web landmarks is shrinking. With the popularity of mobile devices such as smartphones and home Wi-Fi, it provides a new research direction for the acquisition of street-level landmarks. Therefore, how to obtain a large number of terminal IP and its geographical location, and how to expand the source of street level landmark acquisition horizontally is a problem worthy of attention and research.

There is no fusion algorithm specifically for the characteristics of landmarks, and the commonly used methods are relatively simple, the fusion effect is not good. It does not match the characteristics of landmarks. Therefore, to solve the problem of the small number and low coverage of single-source landmarks, it is worth studying to explore the method of landmarks fusion combined with the characteristics of landmarks.

In the design of algorithm, we should not only pursue high precision, but also fully protect the privacy of users, in order to achieve a win-win future for service providers and users. The algorithm or system should provide different precision location services and privacy protection according to the needs of different users, achieve a balance between the complex precision of location and the calculation cost, and get more users' support.

VI. CONCLUSION

Network entity landmark mining technology is an important research direction in the field of network space resource mapping, which has attracted the attention of many researchers at home and abroad. With the continuous development of cloud computing and social networks, the range of application of landmark based IP positioning technology is gradually expanding, and the demand for the street-level landmark is growing. However, with the continuous attention to privacy protection, the acquisition of landmarks has been greatly limited, which brings more challenges to the mining technology of network entity landmarks, and presents more problems that need further exploration and research. I believe that with the continuous efforts of the majority of researchers, the technology will be able to continue to get breakthroughs and make significant progress.

ACKNOWLEDGMENT

This work is supported by the National Nature Science Foundation of China (No. 61572445), NSFC Joint Fund Key Project (No. U1804263).

REFERENCES

[1] Wang Zhanfeng, Feng Jing, Xing Changyou Zhang Guomian, and Xu Bo. "Research on IP positioning technology," Journal of Software, 2014(7):1527-1540.

[2] Venkata N, Padmanabhan, Lakshminarayanan Subramanian, “An Investigation of Geographic Mapping Techniques for Internet Hosts,” ACM SIGCOMM Computer Communication Review, 2001, 31(4):173-185.

[3] Koch R, Golling M, Stiemert L, Rodosek G D, “Using Geolocation for the Strategic Preincident Preparation of an IT Forensics Analysis,” IEEE Systems Journal, 2016, 10(4): 1338-1349.

[4] Zhu Bin, “Research and Implementation of Network Entity Geolocation Technology Based on IP Address,” Beijing University of Posts and Telecommunications, 2015.

[5] Bernard Wong, Ivan Stoyanov, Emin Gun Sirer, “A Comprehensive Framework for the Geolocalization of Internet Hosts,” Octant: Proceedings of the 4th Usenix Symposium on Networked Systems Design & Implementation, 2007: 313-326.

[6] Wang Zhihao, Zhang Weidong, Wen Hui, Zhu Hongsong, Yin Libo, Sun Limin, “Research on IP Location Technology,” Journal of Information Security, 2019,4(03):34-47.

[7] Huffaker, Bradley , M. Fomenkov , and K. Claffy . “DRoP: DNS-based router positioning,” ACM SIGCOMM Computer Communication Review, 2014,44(3):5-13.

[8] Ovidiu Dan, Vaibhav Parikh, Brian D. Davison, “Distributed Reverse DNS Geolocation,” Proceedings of the IEEE International Conference on Big Data. 2018: 1581-1586.

[9] Gueye B, Ziviani A, Crovella M, Fdida S, “Constraint-Based Geolocation of Internet Hosts,” IEEE/ACM Transactions on Networking, 2006, 14(6):1219-1232.

[10] SandorL aki, PeterMatray, P Haga, I Csabai, G Vattayet, “A Model Based Approach for Improving Router Geolocation,” Computer Networks, 2010, 54(9): 1490-1501.

[11] Dong Z, Rohan D.W. Perera, Rajarathnam Chandramouli, K.P. Subbalakshmi, “Network Measurement Based Modeling and Optimization for IP Geolocation,” Computer Networks, 2012, 56(1):85-98.

[12] Ethan Katz-Basstt, John P. John, Arvind K rishnamurthy, “Towards IP Geolocation Using Delay and Topology Measurements,” Proceedings of ACM SIGCOMM Conference on Internet Measurement, 2006: 71-84.

[13] Wang Y, Daniel Burgener, Marcel Flores, “Towards Street-Level Client-Independent IP Geolocation,” Proceeding of USENIX Conference on Networked Systems Design and Implementation, 2011: 365-379.

[14] Jiang H, Liu Y, Jeanna N. Matthews. “IP Geolocation Estimation Using Neural Networks with Stable Landmarks,” Proceedings of the IEEE on Computer Communications Workshops. 2016: 170-175.

[15] Guo C, Liu Y, Shen W, Helen J. Wang， Zhang Y, “Mining the Web and the Internet for Accurate IP Address Geolocations,” IEEE INFOCOM 2009. IEEE, 2009: 2841-2845.

[16] Zhu G, Luo X, Liu F, Chen J, “An Algorithm of City-Level Landmark Mining Based on Internet Forum,” International Conference on Network-Based Information Systems. IEEE, 2015:294-301.

[17] Mun H, Lee Y. “Building IP Geolocation Database from Online Used Market Articles,” Asia-Pacific Network Operations and Management Symposium (APNOMS). IEEE, 2017: 37-41.

[18] Dan O, Parikh V, Davison B D. “Improving IP Geolocation Using Query Logs,” Proceedings of the Ninth ACM International Conference on Web Search and Data Mining. ACM, 2016: 347-356.

[19] Wang T, Xu K, Song J, et al. “An Optimization Method for the Geolocation Databases of Internet Hosts Based on Machine Learning,” Mathematical Problems in Engineering, 2015.

[20] Ma T, Liu F, Zhang F, Luo X, “An Landmark Evaluation Algorithm Based on Router Identification and Delay Measurement,” International Conference on Artificial Intelligence and Security, 2019: 163-177.

[21] Zhu Guang, LI Ke. “Selection of Network Physical Landmarks based on Routing hops,” Journal of Zhongyuan Institute of Technology, 2016,29(01):89-94.

[22] Shavitt Y, Zilberman N. “A Geolocation Databases Study,” IEEE Journal on Selected Areas in Communications, 2011, 29(10): 2044-2056.

[23] Zu S, Luo X, Liu S, Liu Y, Liu F, “City-level IP Geolocation Algorithm Based on PoP Network Topology,” IEEE Access, 2018, 6:64867-64875.

[24] Li H, He Y, Xi R, Wang Z, “A Complete Evaluation of the Chinese IP Geolocation Databases,” International Conference on Intelligent Computation Technology and Automation. IEEE, 2016:13-17.

[25] Liu H, Zhang Y, Zhou Y, “Mining Checkins from Location-S haring Services For Client-Independent IP Geolocation,” IEEE INFOCOM 2014-IEEE Conference on Computer Communications. IEEE, 2014: 619-627.

[26] Zhao F, Shi W, Gan Y, Peng Ziru, Luo X, “A Localization and Tracking Scheme for Target Gangs Based on Big Data of Wi-Fi Locations,” Cluster Computing, 2018: 1-12.

[27] Li R, Liu Y, Qiao Y, Te Ma, Luo X, “Street-Level Landmarks Acquisition Based on SVM Classifiers,” CMC-COMPUTERS M ATERIALS & CONTINUA, 2019, 59(2): 591-606.

[28] Triukose S, Ardon S, Mahanti A, Seth A, “Geolocating IP Addresses in Cellular Data Networks,” Proceedings of the 13th international conference on Passive and Active Measurement, 2012.

[29] Li R, Sun Y, Hu J, “Street-Level Landmark Evaluation Based on Nearest Routers,” Security and Communication Networks, 2018.

[30] Yang W, Liu X, Yin M, “Street-Level Landmark Evaluation with Upper Error Bound,” IEEE Access, 2019, PP(99):1-1.

2020 6th International Symposium on System and Software Reliability (ISSSR)

Wenbo Mi
College of Computer Science and Technology
Xinjiang Normal University
Urumqi, China
mwb874044596@163.com

Yong Li
College of Computer Science and Technology
Xinjiang Normal University
Urumqi, China
liyong@live.com

Shibo Wang
College of Computer Science and Technology
Xinjiang Normal University
Urumqi, China
wangshibo617@163.com

***Abstract*— *Software defect prediction is a popular technical method in software engineering. In order to reduce the cost of a software defects, problems existing in the software are found by testing software products. Software defect prediction often uses machine learning techniques to improve the performance of software testing but requires enough labeled data when training the model. Because the cost of obtaining data is different from the label, the data is easy to obtain, but the label is cumbersome and expensive. In order to demonstrate software defect prediction, after the data obtained active learning algorithm is introduced to query the data, and the most valuable data is selected for expert annotation and then put into the model for training. However, it is not clear which active learning query strategy to choose the most effective in the software defect prediction model. We use different active learning strategy software defect prediction models for comparison. Experiment on the NASA dataset, using Naive Bayes and SVM, Linear Regression as the classifier. Comprehensive research results show that the Density-weighted strategy has a significant effect on the data set.*

Keywords*—*Software defect prediction*, *Active learning*, *Machine learning*, *Query strategy

I. INTRODUCTION

In software engineering, software developers' design of software framework structure and software code errors will have a huge impact on the software system. Therefore, the traditional software testing method is used to test the entire software system. The number of faulty modules is much less than that of non-faulty modules. Traditional software testing methods require a lot of manpower and resources, and work efficiency is very low. According to the statistics report of International Business Machines Corporation. Software testing spends 50%-75% of the total resources of the entire software life development cycle [1]. Traditional software testing is not suitable for products with huge orders of magnitude. Therefore, software defect prediction replaces traditional software testing methods.

Software defect prediction models usually measure the coupling of modules and the complexity of code attributes. Based on the metric data of software prediction, combined with machine learning research methods to build a software defect prediction model [2]. Find the defective module through the predicted model. Staff can spend benefits and resources on modules that contain more defects. Ultimately achieve the goal of optimizing the entire project, improving work efficiency and reducing losses. Traditional machine learning-based software defect prediction commonly uses Naïve Bayesian (NB), Decision Tree, Linear Regression (LR), support vector machines (SVM) and other algorithms to build models for software defect data sets. The software defect prediction model is very sensitive to the feature dimension, measurement information, and annotation content of the defective module. Therefore, high-quality annotation is required to accurately build the prediction model. Software defect prediction faces the following problems. Contains a lot of easily accessible unlabeled data. Its cost is lower, the amount of marked data is small, and it is more difficult to obtain, expensive and costly. Therefore, in view of the problems of traditional software defect prediction in the establishment of data sets. We can use active learning to quickly build data sets.

Angluin[3] first proposed active learning to solve the problem of expensive labeling. You can use fewer labels to train the model and get better results. Therefore, it is very important to train the model with fewer labels. Active learning can filter samples with good classification effect through different query strategies. Then put it into the model through manual annotation. Iterate this process until certain conditions are met. Thereby reducing the pressure in machine learning training and improving efficiency. In many software defect prediction problems, active learning has a good performance. The software defect prediction combined with active learning is a binary classification problem: defective modules and non-defective modules. Predictive models are needed to find defective modules. Commonly used query strategies for active learning are: Uncertainty Sampling, Query-By-Committee, QueryInstanceQUIRE, Density-weighted, QueryRandom. We researched and solved the following problems:

Question1：Among the active learning strategies studied, which strategy can get the best model for software defect prediction?

There are two variables that affect the performance of software defect prediction: (1) classifier type, (2) active learning strategy. Therefore, before solving this problem, it is necessary to study the variables that actively learn software defect prediction performance. In this article, we choose LR, SVM, NB as classifiers, combined with the above five active learning strategies, made an experimental comparison on the NASA data set of real software defect prediction.

Yong Li is the corresponding author (e-mail: liyong@live.com), and also with Key Laboratory of Safety-Critical Software (Nanjing University of Aeronautics and Astronautics), Ministry of Industry and Information Technology, Nanjing, China.

This work was supported by the National Natural Science Foundation of China (Grant No. 61562087, U1703261 and 61662082), the Doctoral Sci entific Research Foundation of XinjiangNormal University (Grant No. XJN UBS1905), and funded by the Scientific Research Program of the Higher E ducation Institution of Xinjiang, China (Grant No. XJEDU2017S031).

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00021

Question2: Compared with traditional software defect prediction methods, how does software defect prediction using active learning strategies perform and how well?

Lu, Xu et al. [5-6] confirmed that active learning strategies can improve the performance of software defect prediction models. Evaluation performance needs to be compared with the experiment. In the third section of this paper, the evaluation performance index will be introduced.

The software defect prediction model is directly related to the active learning query strategy. All of these strategies are sufficient to affect the software defect prediction model. However, the performance of these active learning strategies in the model has not been compared. The contribution of this paper is as follows: Choose five representative query strategies in the software defect prediction model based on active learning. Empirical comparisons are made on the three classifiers. Get the most suitable strategy for software defect prediction. These conclusions will provide valuable research ideas and experimental guidance for data screening and model prediction. The structure of the rest of this article is as follows: In the second part, the five strategies of active learning and applicable scenarios are introduced in detail. The third part introduces the experiment setting: evaluation index, data set, model, experiment design. The fourth part gives the experimental comparison and analysis. Finally, the conclusion is introduced in the fifth part.

II. STRATEGY ALGORITHM

The research of active learning shows that according to different query sample methods, it can be divided into flow-based and pool-based methods. The commonly used strategy is the pool-based strategy [4]. Now it has been involved in many fields: such as image, text classification and so on.

A. Active learning strategy

1) Uncertainty Sampling(Unc)

Uncertainty Sampling is a confidence strategy based on information entropy. Information entropy can be used to measure the amount of information. This strategy uses information entropy to filter the most valuable samples and hand over the selected samples to experts for annotation. Through the least labeling, minimizing the loss function, combined with the machine learning model to achieve the best classification effect. Tong and Kolle[7] et al. combined with the SVM model for research. Use SVM model to find the best hyperplane in high-dimensional space. Then completely separate the defective module from the non-defective module. It is better to use active learning strategies. The calculation formula is:

$$x^* = \arg\max_{i=1\ldots n} -\sum_i p(y_i|x_i)\log p(y_i \mid x_i) \qquad (1)$$

Based on information entropy, it is commonly used to determine the uncertainty in machine learning. In this formula, y_i represents the measurement of information entropy for all unlabeled data. Software defect prediction is a binary classification problem. Using this strategy is more suitable to judge the confidence strategy.

2) Query-By-Committee(QBC)

This strategy is called QBC for short, and it uses the existing labels to construct multiple classifiers, usually called "committees", which have the final decision-making power. Each classifier will make a prediction vote based on the unlabeled samples, and select the sample with the largest difference to be labeled. Only part of the samples can be used to make decision judgments and put them into the training set. Therefore, this method has lower computational complexity and faster running speed. QBC is based on minimizing the version space and reducing this space as much as possible. So the purpose of this strategy is to find the most objectionable space or area, so as to label, and find the optimal solution in the hyperplane space. QBC strategy is similar to the integrated learning in machine learning, and it can show a good effect in multiple models. The two methods commonly used in the QBC method are relative entropy [8] and voting entropy [9] to measure the difference. The calculation formula is:

$$D(x_i) = -\frac{1}{\ln\min(K,C)}\sum_{k=1}^{c}\frac{V(c_k,x_i)}{K}\ln\frac{V(c_k,x_i)}{K} \qquad (2)$$

$$D(x_i) = \frac{1}{K}\sum_{i=1}^{K} D[P_i(C \mid x_i) \,||\, P_{arg}(C|x_i)] \qquad (3)$$

P represents the number of committees and C represents the collection of categories. P represents the number of committees and C represents the collection of categories.

3) QueryInstanceQUIRE(QUI)

The first two active learning algorithms are looking for the most controversial examples. However, some internal relationships among instances may be ignored. Thereby affecting the correctness of classification. There is correlation between instances. There is also a certain data structure between the labeled and unlabeled samples. Therefore, it is particularly important to find such unlabeled examples with the above characteristics. The QUIRE strategy solves the above problems. The measurement evaluates the amount of information of unlabeled instances and their correlation [10]. This method solves the binary classification problem of software defect prediction. It also solves the problem of image processing and text classification with multiple labels and multiple classifications.

4)Density-weighted(Dens)

The first query strategy, Uncertainty Sampling, fuse information entropy to find samples that need to be annotated. This strategy is effective, but when encountering some special circumstances, there will be deviations: the sample contains multiple abnormal points. If the query strategy selects such anomalies, it will greatly affect the results. So using the Density-weighted strategy for sampling can avoid selecting such anomalies.

Through density measurement and density weighting. Combine the k neighbor algorithm to calculate the Euclidean distance and count high-density instances. Generally speaking, a sample with a high-density weight indicates the presence of a large number of such samples. And it will not become an abnormal point of the sample. Therefore, Density-weighted avoids anomalies in the decision plane, thereby optimizing the model of software defect prediction.

5）QueryRandom(Ran)

The QueryRandom strategy is extremely random, and there are no specific conditions for the selection of tags.

Usually, it is compared after using other active learning strategies. Software defect data sets often have conditions that are class-unbalanced and unstable under the performance of this strategy.

B. Classifiers

1 ） SVM is a machine learning method based on structural risk minimization. Its application in software defect prediction is: SVM often encounters linear indivisibility in actual problems. Therefore, after learning, the SVM is mapped to the hyperplane, and the maximum classification interval is divided by the hyperplane. Divide the modules into defective modules and non-defective modules. The interval here refers to the distance between the hyperplane and its closest defective and non-defective modules. So in the learning process, the learning strategy selected by the learner to select unlabeled samples needs to be judged by means of SVM.

2） Linear regression

Linear regression mainly builds a linear regression model by collecting effective software defect prediction data sets. Then calculate the degree of influence of various factors from the model. Usually, the proportional relationship between the independent variable and the dependent variable is studied. The parameters are estimated by the least square method to obtain the predicted software defect module.

3） Naive Bayes

Naive Bayes is a constrained network. It can be used as a stable classifier under the condition of independent software defect prediction class attributes. It is suitable for small-scale data sets such as software defect prediction. It is also insensitive to the data missing in software defects, and it can also stably label the samples in the current test set.

```
Algorithm  Active learning strategy
Input:
        D: Software defect prediction data set of instance
Initialize: Data model for initialization
Require：
                Split dataset into training, testing, labeled, unlabeled set
        Calculate Active learning strategy to find K
Repeat for  k  in  unlabeled set
                Label k
          Put k in labeled
End  for
          invoke traditional and state-of-ML methods
manage your labeled indexes and unlabeled indexes
until the number of queries or the required AUC is reached
```

III. EXPERIMENTAL SETUP

A. *Evaluation indicators*

The central problem studied in this paper is software defect prediction, which is a dichotomy problem. You can use a confusion matrix to make judgments. So as to determine the category predicted by the model and the real sample category after learning. There are four general categories: TP, TF, TN, FN, and TP+FP+TN+FN = total number of samples.

In the software defect prediction problem. Positive examples usually represent defective modules, and negative examples usually represent non-defective modules. Positive examples are the modules we are looking for. Usually the number of such defect-free modules is high, and so the dataset of software defect predictions often exhibits a class imbalance. TP indicates that the real sample is a defective software module. And the classifier also predicts the number of defective software modules, which is obviously the correct classification; FP represents the number of software modules. The real samples of these software modules have no defects. But it is predicted to be defective by the classifier, which is obviously a misclassification; TN indicates that the real sample is a software module without defects. But the classifier is predicted to be a defect-free software module, which is obviously a wrong classification; FN indicates that the real sample is the number of software modules without defects. And the classifier also predicts the number of software modules without defects, which is obviously the correct classification.

TABLE I. CONFUSION MATRIX CLASSIFICATION

The true situation	**Forecast result**	
	Defects	***No Defects***
Defects	TP(true positive)	FP(false positive)
No defects	FN(false negative)	TN(true negative)

The data set used for software defect prediction exhibits unbalanced characteristics. Therefore, evaluation metrics such as accuracy cannot be used. Even if the results of indicators such as accuracy rate perform well, it is meaningless for the class imbalance experiment. So use PD (prediction rate) and PF (false positive rate) in software defect prediction, Draw the ROC curve through PD and PF. The abscissa of the ROC curve is PD and the ordinate is PF. Finally, the area under the ROC curve area is obtained. That is AUC. As shown in “Fig.1 ”.

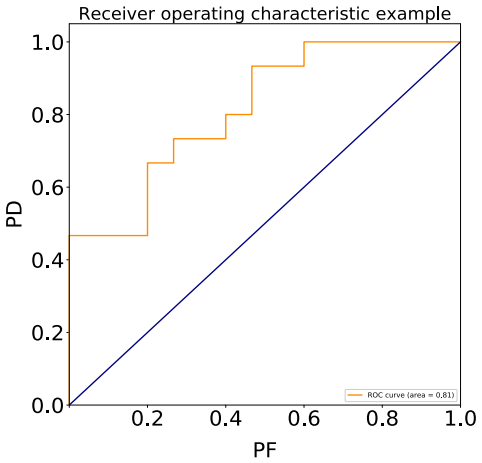


Fig. 1. AUC results

The larger the value of AUC, the better the learning effect of the classifier. The more accurate the model, and the more accurate it can predict defective modules in the software. If the AUC value of one learner is larger than the AUC value of the other learner, the former classifier has better performance. AUC is the best indicator for solving the class imbalance problem, so this experiment uses AUC values to evaluate the model.

B. *Benchmark dataset*

The dataset used in this experiment is from NASA's software system, the NASA dataset. As shown in Table 2.

The NASA dataset is dedicated to software defect prediction models. Contains the static code attribute index of each module, the number of defective modules, and the marking data of software defects.

TABLE II. DATA SET

DATA				
Data	***Language***	***Examples***	***Attributes***	***Defect***
JM1	C	7720	22	1623
KC1	C++	1162	22	294
KC3	Java	1145	21	285
MC1	C++	1152	38	36
MC2	C++	123	39	44
MW1	C	247	37	25
PC1	C	676	37	54
PC2	C	718	36	16
PC3	Java	1050	37	130
PC4	C++	1260	37	176
PC5	C	1692	38	457

C. *Prediction models*

- This experiment uses SVM, Linear regression, and Naive Bayes as the underlying classifier. The well-known literature [11-12] shows that these classifiers are very suitable for software defect prediction and can improve the accuracy of software defect prediction. As a low-level classifier, SVM can use support vectors to define the boundary between defective and non-defective modules in software defect prediction. This type of method is suitable for classification with a small range and has no effect on the size of the sample space. Whether it is linearly separable, inseparable, or nonlinear, it can reflect the decision-making aspect. In order to find the optimal solution of the classifier using the kernel function and the decision function. Linear regression calculates the objective function through sample global information statistics, which is also a commonly used method in software defect prediction. The defect prediction classifier based on Naive Bayes software is based on the independence of features. The characteristics between them are independent of each other. Menzies et al. Log processing is performed according to the static properties of the module, and then a software defect prediction model is established. Finally showed very good results [13]. The Naive Bayes model is superior to the general model, screening for differences between each class, rather than the entire covariance matrix. Therefore, combined with the active learning strategy, the above model is used for experiments.

D. *Experimental design*

In order to explore which active learning strategy has the best effect on the software defect prediction model, an experimental comparison method is used. Therefore, each active learning strategy is combined with a machine learning model for experiments. Active learning strategies will use the method of finding the best labeling examples, rather than the method of batch labeling. First, the experiment normalizes each data set, and divides the data set into a training set and a test set, which are used for the training and testing of the experiment. The data used for the training set is divided into labeled data set and unlabeled data set. They are represented as marked and unmarked in the experiment, and the marking is processed through index management. Initialize the model and set the mark rate of the entire experiment to a fixed value of 10%. Adjust the strategy of active learning, and mark the valuable samples found. Here, the original unread labels replace the manual labeling, and then add the labeled samples to the model for training. Record the AUC value every time, and iterate the whole process until the condition is reached. In this experiment, the number of active learning queries is set to 50. This can better filter the samples and use a 10-fold cross-validation method. The average value of the AUC values obtained each time is used as the final result of this experiment. All experiments use classifiers in Scikit learning. Software defect prediction data has the characteristic of imbalance, so all weight parameters are set to balance [14].

IV. ANALYSIS OF RESULTS

The following are the results of this experiment and the analysis of the results. At the beginning we raised two questions, and the results will be explained below. “Fig 2” shows the experimental results:

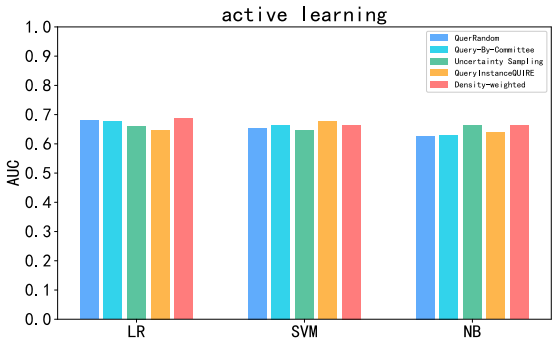


Fig. 2. Strategy diagram

Table 3 shows the AUC values of the prediction model

TABLE III. LR EXPERIMENTAL RESULTS

DATA	LR				
	Ran	***QBC***	***Unc***	***QUI***	***Dens***
JM1	0.632	0.633	0.634	0.627	**0.649**
KC1	0.703	0.666	0.667	0.674	**0.713**
KC3	0.591	0.573	0.556	0.574	**0.601**
MC1	0.693	**0.695**	0.692	0.568	0.664
MC2	0.620	**0.644**	0.628	0.613	0.622
MW1	0.613	**0.706**	0.600	0.653	0.704
PC1	0.668	0.661	0.624	0.663	**0.675**
PC2	**0.649**	0.640	0.610	0.635	0.634
PC3	**0.686**	0.638	0.639	0.674	0.672

PC4	0.741	0.709	0.732	0.734	**0.744**
PC5	0.593	0.890	**0.898**	0.717	0.889

TABLE IV. SVM EXPERIMENTAL RESULTS

DATA	SVM				
	Ran	***QBC***	***Unc***	***QUI***	***Dens***
JM1	0.629	0.627	0.631	0.574	**0.637**
KC1	0.664	0.662	0.666	0.720	**0.694**
KC3	**0.569**	0.567	0.558	0.554	0.567
MC1	0.661	**0.724**	0.602	0.719	0.682
MC2	0.633	0.621	0.604	0.610	**0.636**
MW1	0.600	**0.663**	0.604	0.662	0.632
PC1	0.596	**0.628**	0.599	0.623	0.603
PC2	0.588	0.569	0.608	**0.743**	0.579
PC3	0.647	0.645	0.625	0.658	**0.665**
PC4	0.733	0.712	0.715	**0.774**	0.708
PC5	0.876	0.897	0.893	0.812	**0.909**

TABLE V. NB EXPERIMENTAL RESULTS

DATA	NB				
	Ran	***QBC***	***Unc***	***QUI***	***Dens***
JM1	0.583	**0.602**	0.593	0.541	0.584
KC1	0.676	0.671	0.688	**0.700**	0.684
KC3	0.556	0.540	0.546	0.541	**0.566**
MC1	0.738	0.757	0.732	**0.761**	0.758
MC2	0.632	0.639	0.615	**0.647**	0.633
MW1	0.489	0.510	0.496	0.499	**0.515**
PC1	0.498	0.516	0.521	**0.575**	0.525
PC2	0.502	0.499	0.541	0.499	0.500
PC3	0.558	0.543	0.791	0.633	0.766
PC4	0.792	0.793	0.877	0.776	**0.880**
PC5	0.878	0.878	0.890	0.867	0.871

Question1: Among the active learning strategies studied, which strategy can get the best model for software defect prediction?

For problem 1, through the NASA dataset, five active learning strategies are used to compare the AUC values on the three classifiers. As shown in the table. In the table, each active learning strategy is described in a column. Through 10% of the number of tags, combined with the active learning strategy to form a feature subset, the software defect prediction model is verified. As can be seen in the table, although Random performed well three times, the overall performance was poor. The QBC strategy has only one good performance in the NB model [15]. The Uncertainty strategy did not show a good performance on the SVM model. It shows that the NASA data set contains many anomalies. These abnormal points can affect the classification of the decision plane by the Uncertainty strategy. The QUIRE strategy performs generally in the LR model, but performs well in the SVM and NB models, and it can get the highest value of 0.774 under the PC4 SVM model compared to other strategies. It shows that QUIRE has potential application performance in software defect prediction. It can be combined with other strategies for optimization [16]. The Density strategy is more effective than other active learning strategies, and the effect is obvious under the three machine learning models. In particular, the AUC value of the PC5 SVM model can reach a high performance of 0.909. Density shows strong stability in the field of software defect prediction. The class imbalance is screened by density to determine valuable and high robust examples. “Figures 3,4 and 5” show the model prediction box and whisker diagram.

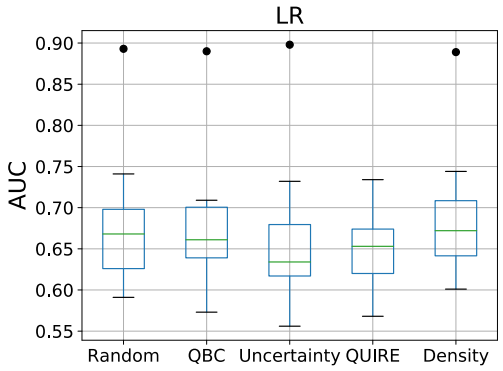


Fig. 3. LR model

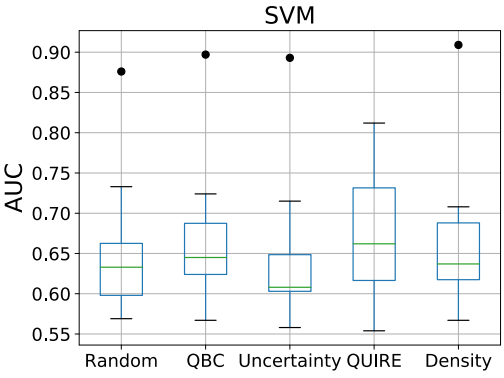


Fig. 4. SVM model

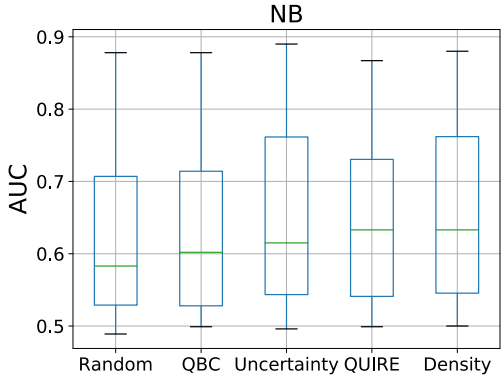


Fig. 5. NB model

Question2: Compared with traditional software defect prediction methods, how to perform software defect prediction using active learning strategy and how is its performance?

Software defect prediction data and prediction models are important research contents of software defect prediction. Traditional software defect prediction is mainly based on a supervised prediction model. There are also some software defect predictions based on semi-supervised and unsupervised classification models. Supervised learning is to learn on labeled examples to improve generalization ability. This work has high requirements for the labeling of the data set. Rish in [17] showed that Naive Bayes can show good performance under a large number of tags and the data meets the independent distribution conditions. If the above conditions are not met, the classification effect will be very poor. And the class imbalance of the software defect prediction data set will also affect the learning of the classifier [18]. Active learning strategies can improve sample quality by screening valuable tags. Then combine the naive Bayes model to make the classifier learn more effectively. Other models can also better predict defective modules through high-precision examples. In general, combined with active learning strategies, higher AUC values can be obtained, thereby better predicting defective modules [19].

V. CONCLUSION

Software defect prediction is an important development direction of artificial intelligence. Continuously optimize in this direction. The experience of active learning in data mining can be known. Combine active learning with software defect prediction to get a better classification effect [20].

This article studies active learning strategies. Summarizes the current five strategies commonly used in the direction of software defect prediction. Combining the three models of machine learning for conceptual discussion and experimental comparison. The research results show that the Density strategy has the best performance on the NASA data set.

Active learning can provide strategies for problems such as larger data, fewer tags, and high cost. However, in the face of many problems in software defect prediction, such as data imbalance, dimensionality disaster, and noise problems, no in-depth research has been conducted, so we can further solve above issues in the future works.

REFERENCES

[1] M. J. Harrold and A. Orso, "Retesting software during development and maintenance," 2008 Frontiers of Software Maintenance, Beijing, 2008, pp. 99-108, doi: 10.1109/FOSM.2008.4659253.

[2] Chen X, Gu Q, Liu WS, Liu SL, Ni C. Survey of static software defect prediction. Ruan Jian Xue Bao/Journal of Software, 2016,27(1):1-25 (in Chinese).

[3] Angluin, D. Queries and Concept Learning. Machine Learning 2, 319–342 (1988).

[4] Burr Settles. Active learning literature survey[R]. Computer Sciences Technical Report 1648. Madison:University of Wisconsin-Madison,2009.

[5] Huihua Lu and Bojan Cukic. 2012. An adaptive approach with active learning in software fault prediction. In Proceedings of the 8th International Conference on Predictive Models in Software Engineering (PROMISE '12). Association for Computing Machinery, NewYork,NY,USA,79–88.

[6] Xu Z , Liu J , Luo X , et al. Cross-version defect prediction via hybrid active learning with kernel principal component analysis[C]// IEEE International Conference on Software Analysis. IEEE, 2018.

[7] Lindenbaum, M., Markovitch, S. & Rusakov, D. Selective Sampling for Nearest Neighbor Classifiers. Machine Learning 54, 125–152 (2004).

[8] Andrew McCallum and Kamal Nigam. 1998. Employing EM and Pool-Based Active Learning for Text Classification. In Proceedings of the Fifteenth International Conference on Machine Learning (ICML '98). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 350–358.

[9] ArgamonEngelson, Shlomo, Dagan, Ido. Committee-based sample selection for probabilistic classifiers.[M]. AI Access Foundation, 1999.

[10] S Huang, R Jin, Z H Zhou. Active learning by querying informative and representative examples[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(10) : 1936-1949.

[11] Menzies T, Greenwald J , Frank A . Data Mining Static Code Attributes to Learn Defect Predictors[J]. IEEE Transactions on Software Engineering, 2007, 33(1):2-13.

[12] S. Lessmann, B. Baesens, C. Mues, and S. Pietsch, ''Benchmarking classification models for software defect prediction: A proposed framework and novel findings,'' IEEE Trans. Softw. Eng., vol. 34, no. 4, pp. 485–496,Jul. 2008.

[13] T. Menzies, J. Greenwald, A. Frank.Data mining static code attributes to learn defect predictors. IEEE Transactions on Software Engineering, 2007, 33(1): 2-13

[14] S. Wang and X. Yao, ''Using class imbalance learning for software defect prediction,'' IEEE Trans. Rel., vol. 62, no. 2, pp. 434–443, Jun. 2013.

[15] Seung, H Sebastian, Manfred Opper, Haim Sompolinsky. Query by COMMITTEE[C]. Proceedings of the fifth annual workshop on Computational learning theory. New York : ACM, 1992.

[16] S Huang, R Jin, Z H Zhou. Active learning by querying informative and representative examples[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(10) : 1936-1949.

[17] I. Rish. An empirical study of the naive Bayes classifier. In: Proceedings of the Workshop on Empirical Methods in Artificial Intel ligence, Seattle, Washington, USA, 2001, 41-46

[18] Li Y, Huang Z, Wang Y, et al. Evaluating Data Filter on Cross-Project Defect Prediction: Comparison and Improvements[J]. IEEE Access. 2017, 5(99): 25646-25656.

[19] Shepperd M, Song Q, Sun Z, et al. Data Quality: Some Comments on the NASA Software Defect Datasets[J]. IEEE Transactions on Software Engineering. 2013, 39(9): 1208-1215.

[20] Shivaji S, Whitehead E J, Akella R, et al. Reducing Features to Improve Code Change-Based Bug Prediction[J]. IEEE Transactions on Software Engineering. 2013, 39(4): 552-569.

A Method of Safe and Fast Bluetooth Connection and Energy Saving for Educational Environment

[1]Jingxian Zhou,[2]Guangming Zheng*,[3]Hui Li,and [1]Zhaojun Gu
[1]*College of Computer Science and Technology, Civil Aviation University of China*
[2]*School of Finance, Nankai University*
[3]*Shijiazhuang Huizhi Technology Co., Ltd*
Tianjin,300350
zhenggm@nankai.edu.cn

***Abstract*—In today's digital era, the combination of digital technology and education to promote education has become a trend. One example is the pen control system. In the current mainstream pen control system, the communications between the smart pen and the smart terminal device like a tablet computer all use Bluetooth technology. This paper designs the self-adaptive identification and pairing scheme between the Bluetooth pen and the smart tablet by analyzing the Bluetooth protocol structure. This paper implements the application software of the Bluetooth master based on the Android tablet,which illustrates the feasibility of the scheme.**

***Index Terms*—Smart Pen,Bluetooth,Auto Pairing,Humanized Interaction**

I. Introduction

In today's digital era, the combination of digital technology and education to promote education has become a trend [1], [2].On the one hand, after thousands of years, writing has been deeply integrated into people's lives as an expression of knowledge and opinions [3]. When people write, people's consciousness is only focusing on the expression of the content, and the actions of writing are completed by the body independently, without the processing of former.On the other hand, the large amount of content produced by writing has shortcomings such as difficulty in retrieval , saving, loss the process of writing but digital technology can make up for these shortcomings [4], [5].The digitizing the Writing track can combine writing and digital technology.At present, in the mainstream writing digital technology, the communication between the smart pen and the digital devices adopts bluetooth technology [6], [7].For example, Anoto [8], and nCode [9],etc., as well as secondary development systems based on these technologies [5].

The security mechanisms used in BR/EDR bluetooth technology have evolved over the course of multiple Core Specifications in three phases: legacy, Secure Simple Pairing,and Secure Connections.But in practice, there are bluetooth devices that use legacy security mechanisms to pair and transfer data.The widespread use of these devices needs attention. In the legacy security mechanism, the PIN code plays an important role in the process of pairing, authentication, encryption, etc [10].This article aims at the specific application scenarios of the connection and interaction based on the bluetooth protocol between the smart pen and the smart tablet terminal,in order to improve the convenience of operation and reduce the manual input of the PIN code to achieve the user's fool-like operation, by analyzing the Android System bluetooth access control, propose solutions.

II. Bluetooth Protocol

The official Bluetooth protocol stack of the Linux system is the Bluez protocol stack, and its architecture is shown in Figure 1. It is composed of three parts: the bottom module, the middle layer and the application layer. Among them, the bottom layer module is realized by hardware, and the middle layer and application layer are realized by software.

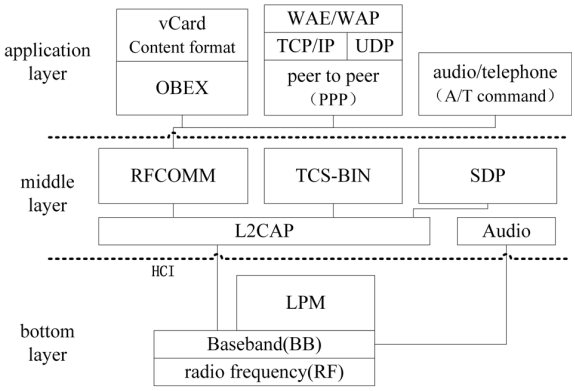


Fig. 1. Bluetooth system protocol structure

A. the bottom module

The Bluetooth bottom module is the core of Bluetooth technology and is an essential part of any Bluetooth device. It is implemented by hardware and is mainly composed of the baseband layer (BB), the connection management layer (LMP) and the radio frequency layer (RF).

The radio frequency layer uses the 2.4GHz ISM frequency band , realizes the filtering and transmission of data streams, and must meet the technical requirements and performance requirements of the Bluetooth receiver.

The baseband layer includes two different physical links (ACL), synchronous connection-oriented link (SCO) and asynchronous connectionless link, and provids circuit switching

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00022

and packet switching technologies. This layer is responsible for the transmission of Bluetooth data frames, and provids three error correction schemes, which are1/3 forward error correction, 3/2 forward error correction and automatic retransmission [11].

The link management layer is responsible for the establishment and removal of Bluetooth device connections,for link security and control, including connection initiation, identity authentication and encryption, negotiating data packet size baseband layer, device power management mode and duty cycle, etc.

HCI (Host Controller Interface), the Bluetooth host control interface, is located between the bottom module and the intermediate protocol. It provides unified commands for calling hardware such as the BB layer, LMP layer, status register, and control register. HCI is the interface between hardware and software in the Bluetooth protocol, and is responsible for interpreting the messages and data,which transfer from the bottom module and the middle layer.

B. the middle layer

The middle protocol layer is a software layer, which is composed of logical link control and adaptation protocol (L2CAP), service discovery protocol (SDP), serial port emulation protocol or cable replacement protocol RFCOM and binary telephone control protocol (TCS).

L2CAP only supports ACL links, works in parallel with LMP, provides connection-oriented and connectionless data services to the upper layer,by using multi-channel technology, segmentation and reassembly technology [12].SDP is a protocol based on the C/S structure, used to query device information and service types,to establish corresponding connections; RFCOMM is a radio frequency communication protocol, a wireless data simulation protocol that emulates a wired link, to support upper layer protocols such as PPP, TCP / IP, etc [11].TCS is a bit-oriented protocol, which defines control commands for establishing voice and data calls between Bluetooth devices.

C. the application layer

The application layer is located in the uppermost part of the Bluetooth protocol stack. It is an optional protocol layer and also a software module. PPP, TCP / IP, and UDP are Internet communication protocols; OBEX protocol is a formatted object exchange protocol, an open standard that defines formats such as business cards, calendars, and memos, that can be used for exchange; WAP is a wireless application protocol used for Digital cellular phones and other small wireless devices to implement Internet services; WAE is a wireless application environment, which provides various application software for WAP phones and personal digital assistants PDA [10].

III. SELF-ADAPTIVE IDENTIFICATION AND PAIRING

The hardware layer of the Bluetooth connection is implemented through the LMP layer, which is regulated and controlled by the software module through the interactive interface provided by HCI. Finally, users can identify the device,authenticate,pairing, and create the link in the operating system or application software.

To ensure communication security, Bluetooth devices need to verify their identity when creating a connection. Bluetooth uses a personal identification number ,also knowed PIN code,as the identity authentication mark, and translates the PIN code into a 128-bit link key for unidirectional and bidirectional authentication. Pairing can be successful only both Bluetooth devices enter the same PIN code [10].

The Bluetooth pen control system proposed in this paper is a non-independent wearable device. Its operating characteristics include inputting on the pen device after creating a connection and responding on the tablet device. If you need to manually enter the PIN code every time you pair, it not only takes time but also affects the convenience of user operations. Through the analysis of the Bluetooth connection process, the software self-adaptive pairing scheme is proposed to cancel the user's PIN code input.

A. Bluetooth connection process

Bluetooth devices are divided into Bluetooth smart devices and Bluetooth smart compatible devices according to different functions.The Bluetooth smart device collects information and sends the information to the Bluetooth smart compatible device; the Bluetooth smart compatible device collects information from the Bluetooth smart device and performs corresponding processing and response procedures [13].

Bluetooth devices are divided into a master and a slave from the perspective of connection: the master actively scans and discovers peripheral devices, identifies the Bluetooth device in a discoverable state, initiates a connection request, and pairs with the target device; the slave is in a discoverable state, passively accepting The connection request ,requires the PIN to verify the identity, and establishes a communication connection with the master after the verification is passed.

In this article, the Bluetooth pen is a Bluetooth smart device and also a slave; the tablet PC is a Bluetooth smart compatible device and a master.

The establishment of the connection of the Bluetooth device needs to go through the steps of opening, querying, pairing and binding the link. In the stage from the start of the Bluetooth device to the successful connection, the Bluetooth device includes the following states:

- Disable state: The master and slave are in an undiscoverable state. The slave cannot be scanned and recognized by any device, nor can the master be found by any other device or scann any other device;
- Enabled state: The master can identify the slave devices in the discoverable state by querying the devices within the working range;
- Discovering state: The master opens the query function to search for Bluetooth devices in the discoverable state in the surrounding work area;

- Discovery Response state: The master finifshes query and returns a list of recognized peripheral Bluetooth devices that can be connected;
- Unbonded state: The Bluetooth device is not paired with the selected target device and establishes a link;
- Bonding state: the master initiates a connection request, the slave requires PIN code identity authentication, the master Bluetooth device sends a PIN code, and the slave verifies;
- Bonded state: After the slave machine passes the PIN code authentication, it establishes a link with the master machine and begins to communicate.

Some states of Bluetooth can coexist, such as the Discovery Response state and Unbonded state. Among them are Disabled state and Enabled state, Discovering state and Discovery Response state, Unbonded state, Bonding state and Bonded state are mutually exclusive. The specific connection state transition process is shown in Figure 2 .

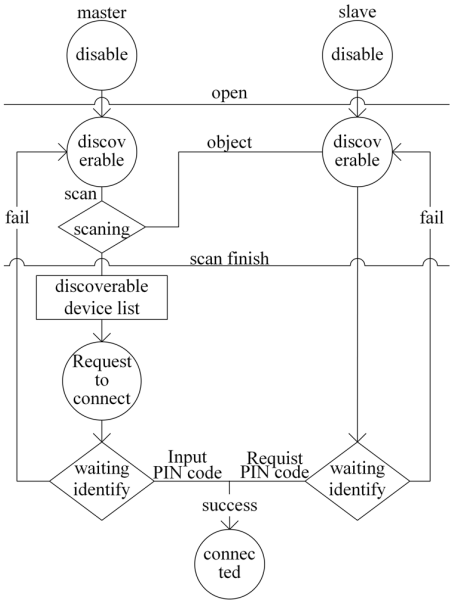


Fig. 2. Bluetooth connection process

B. Self-adaptive PIN code scheme

From the verification stage in Figure 2, if the PIN of the slave is unknown each time it is connected, the master and slave users must negotiate to obtain the PIN code through other communication channels, such as face-to-face, SMS or network communication.Although it can directly obtain the PIN code through the software network communication in application layer to achieve the purpose of automatic pairing. However, this solution is obviously not suitable for wearable Bluetooth devices such as Bluetooth pens with unique functions, simple structure, and no network access module.Therefore, in order to realize user-less PIN code input and Bluetooth smart adaptive pairing, the following three solutions are proposed, as shown in Figure 3.

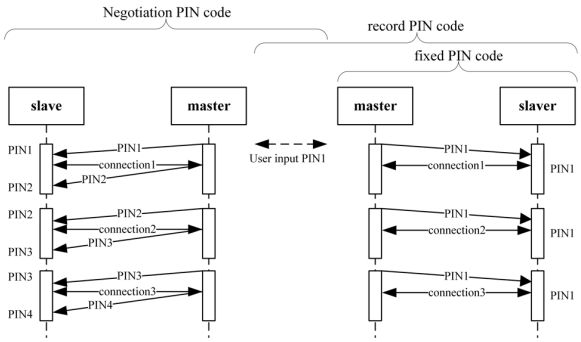


Fig. 3. PIN code self-adaptation scheme

1) Fixed PIN code: Fixed PIN code means that the PIN code of the slave device is fixed to a certain value, such as ”0000”, ”1234”. The value is pre-programmed and stored in the Bluetooth chip.Each time it is paired with this Bluetooth device, the master application software layer directly sends the known PIN code. This method requires the development of application software supporting Bluetooth hardware, which is unique. Therefore, the application of the master can only adapt to the slave Bluetooth device corresponding to the PIN code. Because the PIN code is pre-programmed, it is vulnerable to security attacks, allowing the attacker to obtain the PIN code [14], [15].

2) Record PIN code: The record PIN code scheme is an improved scheme for the shortcomings of the fixed PIN code scheme.In order to improve the scalability of the application and Bluetooth devices with different PIN codes can be flexibly connected, it' a compromise to the user manually enters the PIN code when connecting for the first time.The same as the fixed PIN code scheme, the slave saves the PIN code in the Bluetooth chip or sets it once and does not change it (it is regarded as the master unknown slave PIN code after a change occured). But the master application does not know the PIN code required by the slave when it is connected for the first time, so it still needs to be entered manually.Just as the PC computer uses the MAC address as the unique identification of the device, the Bluetooth device uses the Bluetooth address as the unique identification for establishing the Bluetooth communication link.After entering the PIN code and verifying it successfully, the master records the PIN code and the slave device address in a paire. Then, when the master is connected to the slave device again, it will query from the list of devices that have already passed the verification. If you hit the slave address, you can get the PIN code.In this way, the same effect as the fixed PIN code scheme is achieved. Generally speaking, it is recorded once, and the user never enters again. Recording the PIN code can also strengthen the security of the connection by periodically changing the PIN code.

3) Negotiation PIN code: Negotiation of the PIN code is to further strengthen the security of recording the PIN code, and

the PIN code is set as a dynamic password.The master also does not know the PIN code when it establishes a connection with the slave for the first time, and the user needs to manually enter it after obtaining the PIN code through other channels. After the authentication is successful and the master and slave have established a connection, the master and slave negotiate to determine the PIN code for the next connection, and then both them save the PIN code.When the master and slave are connected next time, the master sends the PIN code negotiated by both parties. This also achieves the effect of a fixed PIN code scheme.However, the PIN code is obtained by the host application software through an algorithm. The PIN code of the slave after the first connection is opaque to the user, and the remaining host devices are also unknown.It is possible to output the latest recorded PIN code on the master with external display output, but this is not applicable to all situations.Then, when the master fails, and the PIN code is lost or when the slave wants to switch to a master, the PIN code record is not transparent, so the pairing connection cannot be made.This requires setting a reset button on the slave side to reset the PIN code to the initial value.In the process of transmitting the PIN code, encryption algorithms such as DES, AES, RSA, etc. can also be added to further enhance communication security.

For the pen control system proposed in this paper, on the one hand, the main transmission data is handwriting data and control commands, in some cases handwriting data is extremely important. On the other hand, it is very inconvenient for the smart pen to modify the PIN code. If the same PIN code is used for a long time, it is no different from the fixed PIN code, which will lead to security problems. Therefore, the pen control system in this article adopts scheme three.

IV. EXPERIMENT

The experiment in this article is based on a pen control system composed of a smart pen and a tablet computer.We use two android tablet to simulate the communication between the smart pen and the tablet computer.

A. Software and hardware environment of the experiment

The software and hardware environment of this experiment is shown in TABLE I. In TABLE I, the Android platform version is the version number of the Huawei M5 tablet's operating system.

B. Implementation of Bluetooth connection pairing

The Bluetooth pairing connection mentioned here refers to the process of the pen control system establishes a secure link through the management of the PIN code.When pairing the smart pen with a master such as a tablet computer for the first time, first reset the PIN code of the smart pen, and then manually enter the PIN code on the master computer side.The master and slave first establish a preliminary secure connection using the manually entered PIN code.After the preliminary secure connection is established, the master application software immediately generates a random PIN code through a random algorithm, and then transmits the random PIN code to the smart pen using the preliminary secure link, and then the pen control system uses the new PIN code to establish a secure link for communication.To prevent the interruption of the link at any time, the PIN code of the next link is unknown. Therefore, the first thing after the establishment of a new secure link is to negotiate the PIN code of the next connection.At this time, the powerful calculate ability of the master is still used to generate a random PIN code through a random algorithm, and the new PIN code is transmitted to the smart pen through a secure link. The smart pen receives the PIN code and stores it locally. At the same time, the master device saves the address of the smart pen and the corresponding PIN code in a local database.The process is shown in Figure 7.

TABLE I
SOFTWARE AND HARDWARE ENVIRONMENT OF THE EXPERIMENT

IDE	Eclipse IDE for Java Developers Version:2018-09 (4.9.0)
ADT	Android Development Tools: 24.2.0.20160729
SDK	android-sdk_r24.4.1-windows
java	java version:1.8.0_162
Android platform version	Android 8.0
Database visualization software	Sqlite expert Personal Edition Version:5.3.1.359(x64)
computer configuration	Intel Core i7-4790 CPU @3.60Hz, 8.0GB RAM, NVIDA GeForce GTX 745
Computer operating system	windows10 pro x64

V. CONCLUSION

Bluetooth has a wide range of application scenarios, and needs to be analyzed and processed differently for specific application environments, operating systems, functional requirements, human-computer interaction, device shape and other characteristics, in order to achieve target of low energy consumption, sound functions, and simplifying humanized interaction.The mobility and openness of the Bluetooth system make Bluetooth widely used, but at the same time make us must strengthen the consideration of security.Although in the legacy security mechanism, PIN code authentication and encryption are provided at the link layer to provide a certain degree of security, it still requires security management at the application layer.After the link is established, it is important to design user data communication protocols with encryption, tamper resistance, and identity verification.

This article takes the convenience of users as the starting point, and based on the consideration of security, it proposes three PIN management methods for Bluetooth self-adaptive identification and pairing.However, the current work is mainly on the application side of the tablet computer, and the design and implementation of the smart pen side are mainly realized by simulation. Therefore, the software and hardware design of the smart pen is the focus of the follow-up work.

VI. ACKNOWLEDGEMENTS

This work is partially supported by the National Key Research and Development Program of China

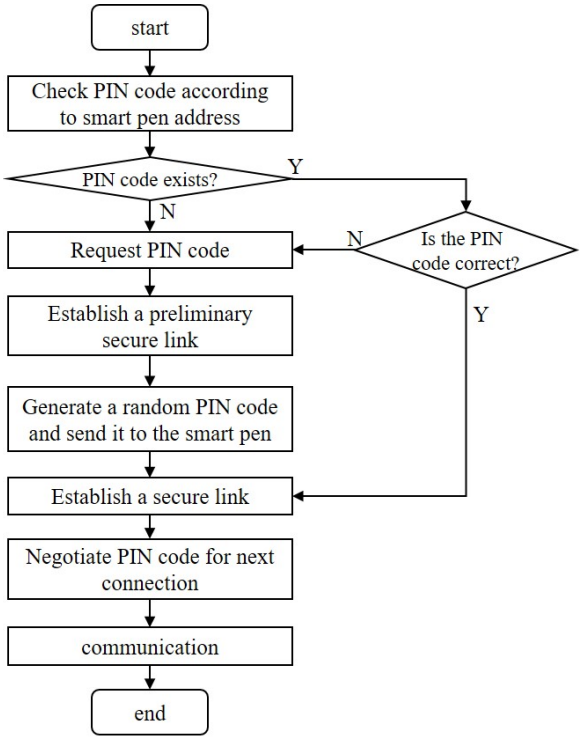


Fig. 4. Flow chart of the master side of the pairing process of the pen control system

(2018YFB2100304), the Natural Science Foundation of Tianjin(19JCQNJC00600,19JCZDJC31600), the National Natural Science Foundation (61872200), the Center Innovation Project (NGII20180306), the Fundamental Research Funds for the Central Universities (Grant No.3122018C036), Project of Shijiazhuang Science and Technology Research and Development Plan (No. 201200024A), and in part by the China Civil Aviation Security Capacity Building Fund Project under Grant PESA 2018082.

REFERENCES

[1] J. E. Hinostroza, C. Labbé, L. López, and H. Iost, "Traditional and emerging it applications for learning," in *International Handbook of Information Technology in Primary and Secondary Education*, J. Voogt and G. Knezek, Eds. Boston, MA: Springer US, 2008, vol. 20, pp. 81–96.

[2] J. Ainley, L. Enger, and D. Searle, "Students in a digital age: Implications of ict for teaching and learning," in *International Handbook of Information Technology in Primary and Secondary Education*, J. Voogt and G. Knezek, Eds. Boston, MA: Springer US, 2008, pp. 63–80.

[3] M. C. Norrie, "The link between paper and information systems," in *Enterprise information systems*, ser. Lecture notes in business information processing, J. Filipe and J. Cordeiro, Eds. Berlin and New York: Springer, 2009, pp. 3–14.

[4] A. Krotoski, "Technology: Libraries of the future," *Nature*, vol. 468, no. 7324, p. 633, 2010.

[5] C. Liao, F. Guimbretiere, K. Hinckley, and J. Hollan, "Papiercraft: A gesture-based command system for interactive paper," *ACM Trans. Comput.-Hum. Interact.*, vol. 14, 2008.

[6] Guo Jianhui, "Research of the dot array based coding and recognition method for digital paper," Ph.D. dissertation, East China Normal University, shanghai(China), 2016. [Online]. Available: http://cdmd.cnki.com.cn/Article/CDMD-10269-1016145761.htm

[7] B. Signer and M. C. Norrie, "Paperpoint: A paper-based presentation and interactive paper prototyping tool," in *Proceedings of the 1st international conference on Tangible and embedded interaction - TEI '07*, B. Ullmer and A. Schmidt, Eds. New York, USA: ACM Press, 2007, pp. 57–64. [Online]. Available: https://doi.org/10.1145/1226969.1226981

[8] Pettersson, Mats Petter and A. Björklund, "Position code," Patent US6 667 695B2, 2003. [Online]. Available: http://www.freepatentsonline.com/6667695.html

[9] LEE, Sang Gyu, MOON, Aram, HAN, Eugene, and PARK, Jiwan, "Electronic pen, electronic device linked," Patent WO2 016 060 409A2, 2016. [Online]. Available: http://www.freepatentsonline.com/WO2016060409.html

[10] A. S. Diallo, W. F. M. Al-Khateeb, R. F. Olanrewaju, and F. Sado, "A secure authentication scheme for bluetooth connection," in *5th International Conference on Computer & Communication Engineering*, T. S. Gunawan, Ed. Los Alamitos, CA: IEEE Computer Society, Conference Publishing Services, 2014, pp. 60–63.

[11] Li Changjiang, "Study on the performance of bluetooth data transmission based on error control," Ph.D. dissertation, College of Communication Engineering, Changchun, 2011. [Online]. Available: http://cdmd.cnki.com.cn/Article/CDMD-10183-1011099827.htm

[12] Z. Mingxing and S. jiao, *Android zhi neng chuan dai she bei kai fa cong ru men dao jing tong*. Bei jing: Zhong guo tie dao chu ban she, 2014.

[13] K. Sairam, N. Gunasekaran, and S. R. Redd, "Bluetooth in wireless communication," *IEEE Communications Magazine*, vol. 40, no. 6, pp. 90–96, 2002.

[14] S. S. Hassan, S. D. Bibon, M. S. Hossain, and M. Atiquzzaman, "Security threats in bluetooth technology," *Computers & Security*, vol. 74, pp. 308–322, 2018.

[15] D. K. Nilsson, P. A. Porras, and E. Jonsson, "How to secure bluetooth-based pico networks," in *Computer Safety, Reliability, and Security*, ser. Lecture Notes in Computer Science, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, F. Saglietti, and N. Oster, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, vol. 4680, pp. 209–223.

Application of NB-IoT Technology in City Open Water Monitoring

[1,2]He Sui,[3]Guangming Zheng*,[1,4]Jingxian Zhou,[5]Hui Li, [1,4]Zhaojun Gu
[1]*Information Security Evaluation Center, Civil Aviation University of China*
[2]*College of Aeronautical Engineering, Civil Aviation University of China*
[3]*School of Finance, Nankai University*
[4]*College of Computer Science and Technology, Civil Aviation University of China*
[5]*Shijiazhuang Huizhi Technology Co., Ltd*
Tianjin,300350
zhenggm@nankai.edu.cn

***Abstract*—Water quality monitoring is an important way to monitor the environment, control pollution and protect water resources, and its research has great practical significance for environmental protection.Aiming at the disadvantages of existing water quality monitoring methods such as high node power consumption and small coverage, this paper proposes an online water quality monitoring system scheme using NB-IOT protocol communication.The prototype system shows that the system can meet the water quality monitoring needs of Bolong Lake.**

***Index Terms*—NB-IoT,lake water quality monitoring,SSM framework,City Open Water**

I. Introduction

On the surface, the proportion of lakes is small, but its role in ecology is very important [1].Especially the lakes in the city, which belong to the City Open Water, are closely related to people's lives.However, increasingly intensive human activities have a profound impact on the health and integrity of aquatic ecosystems [2].In China, freshwater lakes are one of the important sources of water for daily life, industry and agriculture, and the wanton discharge of sewage makes the lake water quality face threats such as excessive heavy metals and eutrophication of water bodies.A total of about 11,000 water environmental incidents have occurred in China since the 1990s, and 60 water pollution incidents have occurred only in 2015 [3]. Eutrophication is one of the ecological disasters that global lakes generally have facing [4].The problem also exists even in developed countries [5].The problems caused by lake eutrophication have forced people to invest a lot of resources to carry out a series of environmental management and ecological restoration projects from the basin to the lake.However, the declining trend of cyanobacteria bloom area caused by lake eutrophication monitored by remote sensing in recent years is not obvious [6].Water quality monitoring is one of the important components of environmental monitoring work.And China has regarded water quality environmental protection as one of the important contents of ecological construction, so real-time monitoring of water quality is particularly important.

Using IoT technology to detect water quality is one of the effective methods. This method detects changes in water quality by placing sensors at the monitoring point, and then transmits the data collected by the sensors back to the server for subsequent analysis through various wireless communication technologies of the Internet of Things [7].This method can realize real-time continuous monitoring of water quality changes. However, this method will have different sensor arrangement density and monitoring range, as well as the frequency of replacement, depending on the wireless communication technology used.

II. Related Work

Pretz once said that IoT (the Internet of things) is that things are connected to the Internet through wireless sensors [8]. There are many cases of studying these technologies and applying them to water quality monitoring or environmental monitoring in foreign countries or in China.Muhammad Ayaz and others studied the routing problem suitable for underwater wireless sensor communication in the ocean [9].Lorena Parra et al. studied low-cost water quality monitoring sensors suitable for fishing grounds [10].Ebrahim Karami et al. proposed the Multisensor Data Fusion wireless network for water quality monitoring [11].B Etikasari et al. designed and developed a water quality monitoring system based on a wireless sensor network deployed on an unmanned aerial vehicle [12].

Narrowband Internet of Things (NB-IoT) is an emerging cellular technology, which belongs to a wide-area low-power network technology. It is transmitted through existing GSM network equipment, and the quality of service is guaranteed [13].City Open Water studied in this article is in the city and has a large water area.So it is in the signal coverage of mobile operators. NB-IoT technology can be applied to City Open Water monitoring and can use existing network infrastructure [14].

III. System Design

A. Features of NB-Iot

Narrowband Internet of Things (NB-IoT) is a new cellular technology introduced in 3GPP Release 13 for providing wide-area coverage for IoT [15].It has many features designed for the needs of the Internet of Things.

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00023

1) excellent coexistence performance with legacy GSM and LTE technologies: NB-IoT uses physical resource blocks in GSM and LTE. According to the position of the NB-IOT carrier relative to the LTE carrier in the spectrum, NB-IoT has three deployment scenarios, namely stand-alone deployment, guard-band deployment, in-band deployment.Fig.1 shows an example of three NB-IoT deployment scenarios [15].For stand-alone deployment,separate frequency bands can be used, GSM and LTE frequency bands can be directly re-cultivation, or existing network idle frequency spectrum can be used, without interference with existing LTE or GSM networks.Guard band deployment is the deployment of NB-IoT network in the guard band at the edge of the spectrum, which can maximize the use of existing spectrum resources, but may interfere with the GSM and LTE system [16].In-band deployment refers to a PRB (Physical Resource Block) resource of the NB-IoT network deployed in the LTE or GSM band.

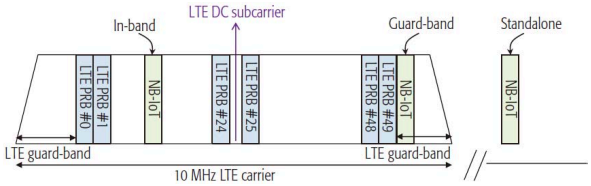


Fig. 1. An example of three NB-IoT deployment scenarios

2) Cover a wide range: NB-IoT achieves a maximum coupling loss 20dB higher than LTE Rel-12 [15], which is 100 times.NB-IoT base station eNB (Evolution Node B) and UE (User Equipment) will select the corresponding information retransmission times according to the CE Level (Coverage Enhancement Level).The criteria for classification are as follows:

- Conventional coverage MCL(Maximum Coupling Loss)$<$144dB, consistent with existing GPRS coverage.
- Extended coverage 144dB$<$MCL$<$154dB, which is an enhancement of 10dB on the basis of existing GPRS.
- Extreme coverage MCL$>$154dB, 20dB enhanced on the existing basis.

3) Low power consumption: NB-IoT has two power saving modes, PSM mode and eDRX mode.PSM mode, namely power saving mode.This mode is equivalent to adding a new state in the original idle state, which is equivalent to shutting down in this state, but still retains the user's context, which can be more convenient when the user enters the idle or connected state.eDRX mode, namely enhanced discontinuous reception. Through the communication between the core network and the user terminal, the terminal can skip most of the paging monitoring and achieve the purpose of reducing power consumption.

These characteristics make NB-IoT extremely suitable for the new water quality monitoring system mentioned in the article. The disadvantage of NB-IoT is that it is not very suitable for connected mobile management, but water quality monitoring does not need to be connected at all times, so its shortcomings can be ignored.The new water quality monitoring system can reduce the energy consumption of sensor nodes through the characteristics of NB-IoT, and is simple to deploy, which is equivalent to an improvement on the traditional wireless water quality monitoring system.

B. Overall system architecture design

In order to realize the overall monitoring of the water quality of City Open Water, sufficient monitoring points must be set up.When deploying water quality monitoring nodes, it is impossible to locate each water quality monitoring point near the monitoring center, and an effective data transmission method needs to be selected.If the data is transmitted by wire, communication cables need to be laid, which is costly and difficult to maintain.The wireless communication does not rely on physical media and has low cost, and can easily realize data transmission between nodes and data convergence centers.City Open Water is located in the city and has sufficient GSM and LTE signal coverage. If the NB-Iot communication protocol is adopted, the existing network facilities can be used for communication.After the water quality information of the monitoring node is transmitted to the monitoring platform, the platform analyzes and processes water quality data. Based on the above analysis, this paper designs a water quality monitoring system consisting of a multi-parameter water quality sensor wireless communication network(MWQSWCN), a water quality monitoring platform(WQMP), and a data display terminal(DDT) to achieve real-time monitoring of the water quality of Bolong Lake. The system architecture is shown in Fig.2.

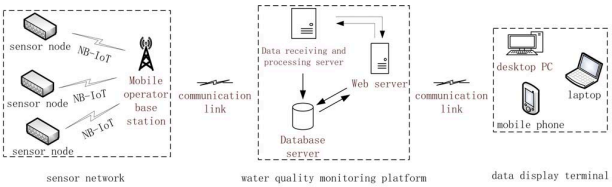


Fig. 2. System architecture diagram

The MWQSWCN is responsible for data collection and forwarding.A sensor node sends the collected water quality data to the operator base station with the strongest signal through the NB-Iot protocol according to the set frequency, and then the operator base station forwards the data to the water quality monitoring platform for subsequent processing and analysis.The water quality monitoring platform is responsible for the reception, processing, storage of water quality parameters, response to user visits, pollution warning, etc.The water quality monitoring platform consists of three parts: a data receiving and processing server, a database server, and a Web server.The data receiving and processing server is the only interface of the water quality monitoring platform to the sensor network. The database server is mainly responsible for storing data, responding to the data receiving and processing server and web server for data insertion, update, and reading operations.The web server is the interface of the water quality

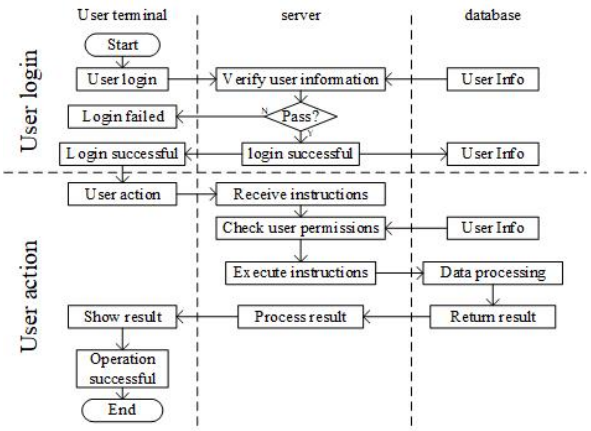


Fig. 3. Web server workflow

monitoring platform to the display terminal. It is responsible for the interaction with the display terminal, the operation of the database information, and the pre-warning information from the data receiving and processing server, and pushing it to the administrator and corresponding users.The workflow is shown in Fig.3.

IV. System Implementation

The City Open Water selected in this article is Bolong Lake.Bolong Lake is located between $39°7'15''\backsim 39°7'48''$north latitude and $117°31'1''\backsim 117°31'29''$ east longitude, and is located in the center of the future science and technology city in Binhai New Area.The water area of Bolong Lake is about 65 hectares, about 1,000 meters from north to south, and about 550 meters at its widest point, as shown in Fig.4.The water area of Bolong Lake is relatively large and the water area is relatively open.Based on this, a water quality monitoring system is implemented.

Fig. 4. Bolong Lake

The server of the water quality monitoring platform is deployed on the Alibaba Cloud server, the operating system is Windows Server 2012, and the platform operating environment is built using Mysql 5.7, Java JDK 8, Tomcat8 and other related software.The data display module can access the Web server through a browser, and also realizes the display of data and the water quality pre-warning on the mobile device.

The Web server is mainly based on the SSM (Spring+SpringMVC+MyBatis) framework and applies Baidu Map API (Application Programming Interface) to realize the visualization of sensor nodes.Mainly composed of the following modules: map display module, equipment management module, water quality data module, etc.Fig.5 is the rendering of the map display module. The water area on the map is Bolong Lake, and the red mark indicates the location of the sensor node.The user can call the device management module by clicking the red mark to view the corresponding device information, such as latitude and longitude, depth, and device code (MAC address). Fig.6

Fig. 5. the rendering of the map display module

Device ID	longitude	latitude	depth	MAC
1	117.53453	39.132959	10.2	0123432-12
2	117.535033	39.129293	15.3	0234623-32
3	117.531134	39.129097	22	3453451-66
4	117.533218	39.13065	12.8	4328342-98
5	117.535958	39.134047	18.1	3218347-54

Fig. 6. device information in the form of list

shows the device management module displaying device information. The system realizes the display of informations of all nodes in a list. Fig.7 is the rendering of the water quality data displayed in the form of a list.

Through the prototype system implemented above, the feasibility of sensor nodes communication based on NB-IoT is verified.

Device ID	PH	Oxygen	Rate	Time
1	8.0	0.40966E-10	11	2018-04-18 17:06:06
1	8.0	0.0540547	11	2018-04-18 17:06:06
1	8.0	0.0532458	11	2018-04-18 17:06:06
2	13.1679	0.685724	21	2017-12-29 11:22:10
3	11.4923	0.697412	21	2017-11-25 10:57:30
4	6.63999	0.495914	0	2017-12-19 09:39:38
5	13.5094	0.495708	20	2017-12-27 18:10:28

Fig. 7. the water quality data displayed in the form of list

V. Conclusion

Aiming at the shortcomings of existing water quality monitoring methods, such as few data collection points, large power consumption, and small coverage, this paper proposes an online water quality monitoring system scheme using NB-IOT protocol to collect nodes data, and implements that through a web browser to Visit the water quality monitoring platform. The prototype system realizes the functions of node online status query, water quality data query.

However, the system still has the following work can be done: 1. The prototype system has a small number of sensor nodes, which leads to a coarser granularity of lake water detection. Second, the prototype system only realized the basic functions,the automatic assessment of water quality has not been realized. These refinements will be future work.

Acknowledgment

This work is partially supported by the National Key Research and Development Program of China(2018YFB2100304), the Natural Science Foundation of Tianji(19JCQNJC00600), and in part by the China Civil Aviation Security Capacity Building Fund Project under Grant PESA 2018082, the CERNET Innovation Project(NGII20180306), the Fundamental Research Funds for the Central Universities of Civil Aviation University of China (Grant No.3122018C036, Grant No. 3122019072), and Project of Shijiazhuang Science and Technology Research and Development Plan(No. 201200024A).

References

[1] X. Chen and L. Feng. Remote sensing of lakes' water environment. *Comprehensive Remote Sensing*, 8:249–277, 2018.

[2] Leanne Elchyshyn, Jean-Olivier Goyette, Émilie Saulnier-Talbot, Roxane Maranger, Christian Nozais, Christopher T. Solomon, and Irene Gregory-Eaves. Quantifying the effects of hydrological changes on long-term water quality trends in temperate reservoirs: insights from a multi-scale, paleolimnological study. *Journal of Paleolimnology*, 60(3):361–379, 2018.

[3] DING Fan, HUANG Li-yong, WANG Rui, G. A.O. Yong-jun, Y. A.O. Jian-yi, WANG Xiao-ye, and L. I. Qun. Water pollution emergencies in china,2004-2015: monitoring data analysis. *Chinese Journal of Public Health*, 33(1), 2017.

[4] Z. H.U. Guangwei, Q. I.N. Boqiang, ZHANG Yunlin, X. U. Hai, Z. H.U. Mengyuan, YANG Hongwei, L. I. Kuanyi, M. I.N. Shen, SHEN Ruijie, and ZHONG Chunni. Variation and driving factors of nutrients and chlorophyll-a concentrations in northern region of lake taihu, china, 2005-2017. *Journal of Lake Sciences*, 30(2):279–295, 2018.

[5] Daniel J. Conley, Hans W. Paerl, Robert W. Howarth, Donald F. Boesch, Sybil P. Seitzinger, Karl E. Havens, Christiane Lancelot, and Gene E. Likens. Ecology. controlling eutrophication: nitrogen and phosphorus. *Science (New York, N.Y.)*, 323(5917):1014–1015, 2009.

[6] Kun Shi, Yunlin Zhang, Yongqiang Zhou, Xiaohan Liu, Guangwei Zhu, Boqiang Qin, and Guang Gao. Long-term modis observations of cyanobacterial dynamics in lake taihu: Responses to nutrient enrichment and meteorological factors. *Scientific reports*, 7:40326, 2017.

[7] Li Chen. *The Node Design of Environment Monitoring Wireless Sensor Network Based on Zigbee*. Master, Shanghai Jiao Tong University, Shanghai, 2008.

[8] K. Pretz. The next evolution of the internet. *IEEE Magazine the Institute*, 2013.

[9] Muhammad Ayaz, Imran Baig, Azween Abdullah, and Ibrahima Faye. A survey on routing techniques in underwater wireless sensor networks. *Journal of Network and Computer Applications*, 34(6):1908–1927, 2011.

[10] Lorena Parra, Sandra Sendra, Laura García, and Jaime Lloret. Design and deployment of low-cost sensors for monitoring the water quality and fish behavior in aquaculture tanks during the feeding process. *Sensors (Basel, Switzerland)*, 18(3), 2018.

[11] Ebrahim Karami, Francis M. Bui, and Ha H. Nguyen. Multisensor data fusion for water quality monitoring using wireless sensor networks. pages 80–85, 2012.

[12] B. Etikasari, Husin, S. Kautsar, H. Y. Riskiawan, and D. P. S. Setyohadi. Wireless sensor network development in unmanned aerial vehicle (uav) for water quality monitoring system. *IOP Conference Series: Earth and Environmental Science*, 411:012061, 2020.

[13] Xingqin Lin, Ansuman Adhikary, and Y.-P. Eric Wang. Random access preamble design and detection for 3gpp narrowband iot systems. *IEEE Wireless Communications Letters*, 5(6):640–643, 2016.

[14] Rashmi Sharan Sinha, Yiqiao Wei, and Seung-Hoon Hwang. A survey on lpwa technology: Lora and nb-iot. *ICT Express*, 3(1):14–21, 2017.

[15] Y.-P. Eric Wang, Xingqin Lin, Ansuman Adhikary, Asbjorn Grovlen, Yutao Sui, Yufei Blankenship, Johan Bergman, and Hazhir S. Razaghi. A primer on 3gpp narrowband internet of things. *IEEE Communications Magazine*, 55(3):117–123, 2017.

[16] Olof Liberg, Mårten Sundberg, Y.-P. Eric Wang, Johan Bergman, and Joachim Sachs. World-class standards. In *Cellular Internet of Things*, pages 15–30. Elsevier, 2018.

Design and Implementation of Intelligent Heart Rate Detection System based on STM32

Zengyu Cai
Zhengzhou University of Light Industry
School of Computer and Communication Engineering,
Henan Key Laboratory of Food Safety Data Intelligence
Zhengzhou, China
mailczy@163.com

Zhongyuan Peng
Zhengzhou University of Light Industry
School of Computer and Communication Engineering,
Henan Key Laboratory of Food Safety Data Intelligence
Zhengzhou, China
869129712@qq.com

Jianwei Zhang
Zhengzhou University of Light Industry
Software Engineering College, Henan Key Laboratory of Food Safety Data Intelligence
Zhengzhou, China
mailzjw@163.com

Yuan Feng
Zhengzhou University of Light Industry
Software Engineering College, Henan Key Laboratory of Food Safety Data Intelligence
Zhengzhou, China
mailfengy@163.com

***Abstract*—At present, people's lifestyle is more and more diverse, and people's eating habits are constantly changing. In this case, people's health is facing great challenges. The sudden rate of various heart rate chronic diseases is also rising. Under the heart rate detection, it can be prevented in advance so that the disease can be effectively treated. This paper designs an intelligent heart rate detection system, which can real-time detect its own heart rate. When the heart rate beats too fast, the voice prompt will be given, and every time the heart rate value detected will be sent to the family through wireless transmission. The heart rate system satisfies the requirements of heart rate LCD display, voice prompt and Bluetooth transmission. It has certain application value for medical treatment.**

Keywords-heart rate, Detection, Voice prompt, Bluetooth transmission, STM32

I. INTRODUCTION

With the rapid economic development, many people have to make changes in their eating habits, want to make life better, do not focus on the combination of work and rest, and overburden the body, which often makes the accumulation of labor and illness, physical health is particularly important in addition, many cardiovascular diseases threaten the body, such as: coronary heart disease causes arteriosclerosis, hypertension, endocrine heart disease and other problems. Early detection of heart diseases is of great significance in medical treatment.

With people's attention to life and health, a large number of research literatures [1-3] on heart rate detection have appeared. Compared with invasive heart rate detection technology, non-invasive heart rate detection based on electronic technology is convenient, safe and efficient. Literature [3] introduces a new non-invasive wrist pulse detection technology, which USES rf sensors to detect heart rate during sleep. Literature [5] introduces a real-time heart rate variability measurement system based on single chip microcomputer, through the adoption of the R - peak detection algorithm, implements the heart rate variability of real-time detection. In literature [6], a photo transmitter and detector through fingertip known as photoplethysmography(PPG) were proposed, and the collected data was transmitted to the Internet. Literature [7], Heart rate variability (HRV) analysis was performed with DSP and for myocardial infarction detection. Literature [8] and [9] introduced the use of fpga to complete heart rate detection. Literature [8] describes a field programmable gate array (FPGA) implementation of a system that calculates the heart rate from Electrocardiogram (ECG) signal. After heart rate calculation, tachycardia, bradycardia or normal heart rate can easily be detected. In literature [9], a non-invasive heart rate monitoring system based on FPGA was proposed for fetal abnormality detection. This heart rate detection technology has the disadvantage of high cost and large volume, so a low cost, small volume and high efficiency heart rate detection system is needed.

In this paper, a simple, practical and low-cost electronic heart rate detection instrument is designed by using electronic technology. This instrument USES single chip STM32F103C8T6 to control the whole system, the heart rate value is displayed, at the same time the voice prompts the current heart rate value, and then through wireless transmission and other functions. The system has the advantages of multi-function, small size, low power consumption and low cost.

II. RELATED TECHNIQUES FOR HEART RATE MEASUREMENT

A. Piezoelectric

Piezoelectric transducer: The pulse of human body is converted into electrical signal by pressure sensor. Pressure ceramics is a very common type of pressure sensor and if you apply it to a piezoelectric element with very high internal resistance, to make an electrode, you have to use two identical elements inside, so that the polarity is reversed, you add a piece of copper between the two identical elements, and you can do

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00024

that. Many sensors are applied in this way, so that the signal output of the sensor is also very large, and the subsequent amplification signal circuit design, the requirements are not high. The price is low, the material is simple, the sensitivity is easy to improve and so on.

B. Acoustic electric

Acoustic electric sensor: Converts mechanical vibrations that travel in blood-soluble gases, body fluids, or bones in skin tissues into electrical signals. Generally, it is a capacitor-type electret microphone. When the electret film receives the sound wave vibration, it will cause the capacitance to change, and then the voltage value will be changed. After the AD conversion, the sensitivity is also very high, but there will be a large signal-to-noise ratio, and the process structure is complex.

C. photoelectric

Photoelectric sensor: The flow of blood in the body can be detected by photoelectric sensors, but rather as a result of changes in light penetration or reflectivity, as a result of the beating of the heart to the rest of the body. The corresponding method can be used to convert the optical signal into the electrical signal. Electrical sensors are highly sensitive to changes in light energy, especially for measuring displacement and distance.

III. Heart rate detection system design

At present, the existing heart-rate meters on the market are generally large and not easy to carry, combined with the market requirements, to create a portable, small size, light weight, easy to carry, of course, must meet the equipment to the heart rate data collection, performance to have accuracy and timeliness. Systematic design is generally divided into two parts: hardware design and software design.

A. Function design

The heart rate detection system can complete the heart rate detection, display the obtained heart rate value, and read it through the voice. When the target person's heart rate is too high, the voice alarm will prompt the heart rate is too high, and the wireless data of the detected heart rate will be transmitted to the mobile phone terminal. Its functionality is shown in Figure 1.

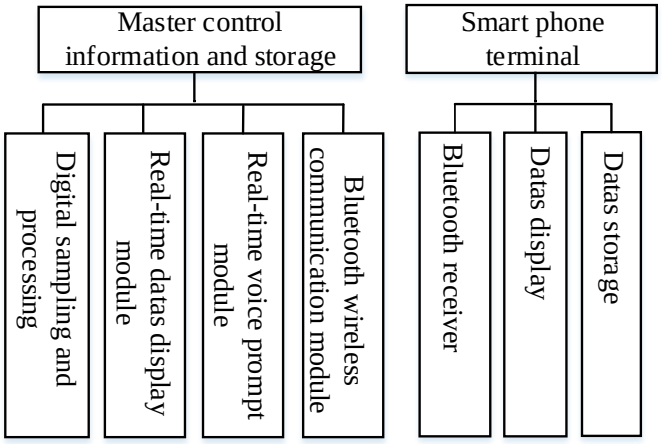


Figure 1. System function

The acquisition terminal of the main control information processing and storage module samples, filters and calculates the heart rate of the human body, and transmits it to other modules through the main control. The display terminal displays the heart rate data in real time, the voice prompts the heart rate data in real time, and the Bluetooth communication transmits the data wirelessly. Mobile phone terminal is mainly used for wireless data reception, display and storage.

B. Structural design of acquisition system

The system collects the heart rate value through the portable heart rate meter, and connects with friends and relatives through the mobile phone client to share the heart rate value, so that family members can check the real-time information of their own heart rate. The whole structure is shown in Figure 2.

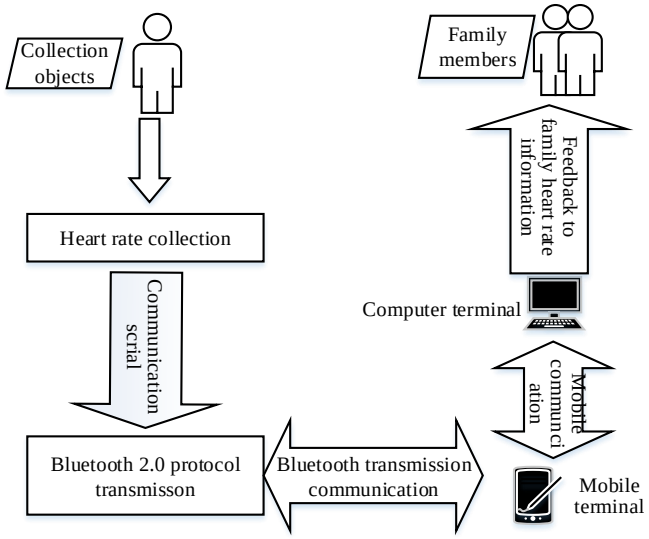


Figure 2. System Architecture Evolution

C. Design of heart rate collection method

The acquisition part USES a reflective photoelectric sensor, a biological sensor module MAX30102 with heart rate monitoring function, a red LED and an infrared LED are integrated internally, the optical detector and optical devices and other components of the module part of the circuit, the electronic circuit can suppress the low noise of the ambient light. This heart rate module adopts a 1.8v logic power supply and a free-standing 5V internal LED power supply, which can be placed in the finger, wrist and other places for measurement. Communication is through a standard i2c compatible interface. The module can be shut down by software with zero standby current to keep the power track supplied. The circuit of heart rate acquisition module is shown in Figure 3

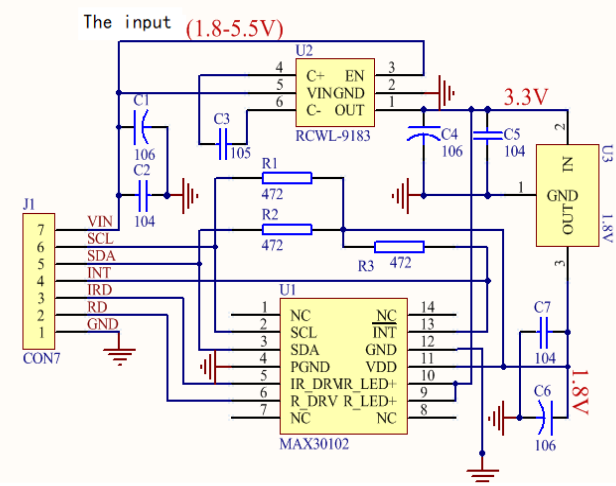


Figure 3. Heart rate acquisition module circuit

D. *Master control chip design*

The main chip is the heart of the system and the most important thing for the whole system. For this design USES the ARM STM32F103C8T6 chip, mainly with the ARM kernel architecture (M3, pure Thumb2 instruction is the use of a processor to carry out, and 8-bit and 16-bit processor series code storage density, a 32-bit high-performance ARM kernel also can be achieved, the ARM MCU development and application of obvious around the world, but the processor core is to design a small volume, light weight, lower price, high reliability, less consumption, not only implements these, also provides many advantages such as flexible interrupt capabilities. In various interrupts and exceptions, increase the speed of corresponding and switching.

E. *Design of language hints*

SYN6288 voice chip is used. SYN6288 communication mode is through asynchronous serial port (UART) communication, in the serial port to the processor, received the text data, after the processor processing, text data to speech or TTS speech conversion. SYN6288 in identify types of text data more accurate, more rapid, high recognition rate, synthetic quality more human voice module, lead with the same price products for heart rate reading needs prompt, based on the system performance analysis, no longer simple to use buzzer, chose SYN6288 voice module, its use maximum play to the integrity of the system and to make clear in any case the heart rate value, can also play a warning role. The circuit of the language prompt module is shown in Figure 4.

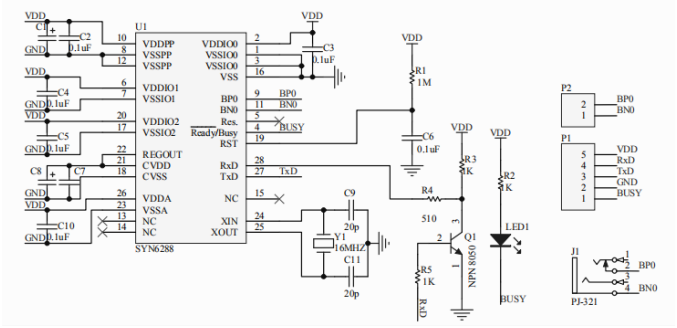


Figure 4. SYN6288

F. *Display module design*

The display module USES the LCD screen, which is a liquid crystal display screen, mainly made of sodium-free glass material and liquid crystal solution. Two pieces of sodium-free glass material are used to place the liquid crystal solution in the middle. The capacitive-loaded components used in LCD screens have very low internal resistance and can be used under both positive and negative pressures, mainly because of their non-polar nature. LCD screen can display a lot of content, for the temperature requirements are very high, too high temperature will make the life decline, the standard service life is very long, almost zero radiation, very low power consumption, small volume and other advantages.

G. *Design of communication mode*

The communication mode adopts Bluetooth wireless transmission mode, with fast transmission rate, low power consumption and high confidentiality. Meanwhile, it manages data and sound transmission, and works under 2.4 GHz frequency. The communication range is about 10 meters.

III. SYSTEM MAIN MODULE SOFTWARE DESIGN

A. *Data collection*

B. If the data is collected to the length of a certain number of 500 samples, it will be judged whether the length of the data of the sample number is enough for 5 seconds. If it is enough for 5 seconds and the sample data is stored, it will not be enough to collect the data again. The frame data is transmitted to the processor to determine whether the data frame is complete. The obtained frame data is completely analyzed. If the frame data is incomplete, the frame data is sent again, waiting for the arrival of the data. Process the parsed data. The details are shown in Figure 5.

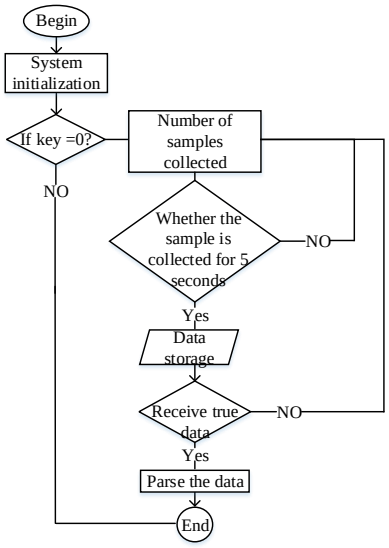


Figure 5. Heart rate collection work

C. Voice prompt

On the power supply, the system into the initial, when key KEY1 press, voice prompt on the collection, collection, when the heart rate information collected, the system according to the algorithm to calculate the heart rate value, and determine whether the heart rate value more than the alert value, and the transmission of data through a serial port, synthesis of text data, call a subroutine to run synthetic voice, speech will read out the current detection of heart rate value, as shown in Figure 6.

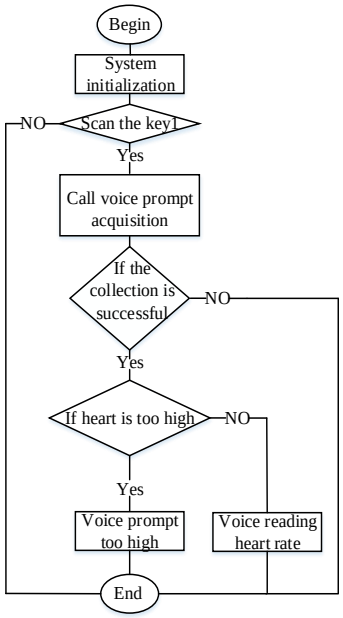


Figure 6. Voice prompt process

V. Test results and analysis

The main content of software testing is to check whether the test software can perform the required functions perfectly. Mainly to test the stability of the overall system, the system to various conditions, to verify whether the function can work normally. There are many factors affecting people's heart rate. In order to make a clear comparison, the test conditions are limited to a certain range for performance test, gender, height, weight and age, and then compared with the market of Samsung smart bracelet. There are two test subjects: test subject A, gender: male, 170 meters tall, 27 years old, weight: 70KG; Test subject B, female, 160 meters tall, 37 years old, 60KG weight. The result is shown in Table 1.

According to the Table 1: considering the influence of height, age, gender and weight, two persons were tested respectively. In order to compare the heart rate value collected by the heart rate detection system and observe the accuracy of the actual heart rate with the heart rate value measured by the measuring equipment, it can be seen that the difference rate the measured value of the heart rate module is 1.03% at the highest and the difference rate the measured value of the heart rate is *0* at the lowest. In comparison with the actual heart rate measurement, the difference between the mainstream heart-rate meter Samsung band and the actual heart rate measurement is 2.06% at the highest and 0 at the lowest. However, the difference rate the six groups of heart-rate data appears, and there are three groups of deviation values, so compared with the mainstream market equipment Samsung hand, the heart-rate system detection is more advantageous.

TABLE 1. TEST RESULT

Test Status	Test object	Actual heart rate	Samsung band results	Difference rate of Samsung band	Ours detects the value	Difference rate of Ours
Tranquillization	A	66	67	1.52%	66	0
	B	62	62	0	62	0
Walk for 5 minutes	A	97	99	2.06%	98	1.03%
	B	94	94	0	94	0
Jog for 5 minutes	A	132	133	0.75%	131	0.75%
	B	128	128	0	128	0

VI. Conclusion

This paper is to design a heart rate detection system, by ana lyzing the system function and the request, around this topic ha s carried on the design, The main functions of the system are he art rate data, main control chip, LCD and voice prompt, Blueto oth communication, etc. in addition to these functional require ments, many basic requirements are described, such as low po wer consumption, low cost, small size .The next step of this pa per is to further improve the accuracy of heart rate detection, so as to make the measurement more accurate and meet people's needs.

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China (No.61672471), Key Technologies R&D Program of Henan Province (202102210176) and Liaoning Provincial Department of Education Project (Grant No. LQN201912).

REFERENCES

[1] Ronald D. Berger, Solange Akselrod, David Gordon. An Efficient Algorithm for Spectral Analysis of Heart Rate Variability[J]. IEEE Transactions on Biomedical Engineering, 1986, bme-33(9):900-904.

[2] Ferrario M , Signorini M G , Magenes G , et al. Comparison of Entropy-Based Regularity Estimators: Application to the Fetal Heart Rate Signal for the Identification of Fetal Distress[J]. IEEE Transactions on Biomedical Engineering, 2006, 53(1):119-125.

[3] Kim S W , Choi S B , An Y J , et al. Heart Rate Detection during Sleep Using a Flexible RF Resonator and Injection-Locked PLL Sensor[J]. IEEE Transactions on Biomedical Engineering, 2015, 62(11):1-1.

[4] Pearson M , Faust O . HEART-RATE BASED SLEEP APNEA DETECTION USING ARDUINO[J]. Journalof Mechanics in Medicine and Biology, 2019.

[5] Wei,Ying-Chieh, Wei, Ying-Yu, Chang, Kai-Hsiung. DESIGN OF A MICROCONTROLLER-BASED REAL-TIME HEART RATE VARIABILITY MEASUREMENT SYSTEM USING A LOW-COMPLEXITY R-PEAK DETECTION ALGORITHM [J]. Instrumentation Science & Technology, 41(3):274-289.

[6] Uwamahoro Raphael, Mushikiwabeza Alexie, Minani Gerard. Universal design of a microcontroller and IoT system to detect the heart rate[J]. IOP Conference Series: Materials Science and Engineering, 263:052037-.

[7] Zakaria, F.; Khalil, M., "Heart rate variability (HRV) analysis using DSP for the detection of myocardial infarction," Advances in Computational Tools for Engineering Applications (ACTEA), 2012 2nd International Conference on , vol., no., pp.15,19, 12-15 Dec. 2012.

[8] Panigrahy D , Rakshit M , Sahu P K . FPGA Implementation of Heart Rate Monitoring System[J]. Journal of Medical Systems, 2016, 40(3):49.

[9] Rasu R, Sundaram P S, Santhiyakumari N. FPGA based non-invasive heart rate monitoring system for detecting abnormalities in Fetal[C]// 2015.

2020 6th International Symposium on System and Software Reliability (ISSSR)

Routing Communication Protocol based on Reliable Active Node

Yu-ping Li
School of Information Technology
Shangqiu Normal University
Shangqiu Henan
ping032200@163.com

Ying Li
School of Information Technology
Shangqiu Normal University
Shangqiu Henan
linda4736@126.com

Yuexin Wang
School of Information Technology
Shangqiu Normal University
Shangqiu Henan
1974994920@qq.com

Abstract—When using traditional OSPF routing protocol to forward packets, network nodes do not participate in reliability work, which leads to low packet forwarding rate, long average end-to-end delay, high energy consumption and short lifetime of network nodes. In order to solve these problems, a high-performance multi-mobile node routing communication protocol based on reliable active node is proposed. According to the functional requirements of the model, the reliable active node is divided into forward active node and reliable active node. Through the estimation of packets transmission delay time, the independence of communication link quality, the calculation of node forwarding moderate index and the selection of active network neighbor nodes, the routing protocol is designed. Simulation results show that packets forwarding rate is improved effectively, the overall energy consumption of network nodes is reduced, the average end-to-end delay is shortened, and the network node lifetime is prolonged.

Keywords—High Performance; Router; Communication Protocol; Reliable Active Node

1 Introduction

As the scope of Internet application continues to expand, a large number of New applications such as information release, distance learning, video conference, etc. These new applications have a common feature that is one to many or many to many reliable data communication transport. It requires the data sender to transmit the valuable data to the receiver efficiently [1,2]. In order to enable these new applications to achieve efficient communication and transmission of data, multicast communication technology has come into people's attention.

However, there are some disadvantages in multicast communication, such as the explosion of feedback information, the serious loss of data and the difficulty of recovering the lost data. A hierarchical communication routing protocol based on Grover quantum search algorithm is proposed by Fan Xincan [3]. Based on the establishment of quantum wireless communication network model and communication channel model, Grover is adopted The quantum search algorithm searches the maximum routing metric within the limited hop number of quantum wireless communication network as the layered communication target path, which avoids the channel conflict in the communication transmission process, and realizes the stable data transmission in the quantum wireless network. However, the routing protocol has the problem of low packet forwarding rate. A hierarchical communication routing protocol based on partition is proposed by Zhai Chunjie [4]. The network nodes are divided into clusters based on the partition method. The cluster head nodes of data transmission are selected by three-level cluster head selection strategy, and the residual energy of network nodes is fully considered to ensure the energy consumption balance of nodes in the cluster [5,6]. However, this routing protocol has the disadvantages of long average end-to-end delay and high energy consumption. A layered communication routing protocol based on weighted threshold is proposed. Multiple disjoint communication transmission paths are constructed by means of hierarchical network security routing. On this basis, multiple communication paths are selected according to the security requirements of the network base station. At the same time,

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00025

the communication data is partitioned by using the weighted threshold, and the communication data is forwarded along the multi-path. However, the routing protocol will shorten the lifetime of network nodes [7,8,9]. Therefore, a high performance router multi-mobile node communication protocol based on reliable active nodes is proposed and designed.

2 Multi mobile node communication protocol

2.1 Layered model of multi mobile node communication protocol

Active network is a kind of programmable network, which can realize the priority configuration of network services. It is one of the most effective technologies to solve the problem of multicast communication. Active network can not only make active nodes participate in reliability work, but also improve data transmission efficiency by customizing specific service functions of active nodes. As shown in Figure 1, the hierarchical model of multi mobile node routing communication protocol in active network is given, which lays the foundation for the subsequent implementation of active network packet communication routing.

From Figure 1, the hierarchical model of multi mobile node routing communication protocol in active network follows the IP layer in Internet network. But the IP layer mainly realizes the whole network interconnection, which cannot guarantee the reliability of communication data transmission. The main function of the active network multicast communication layer is similar to that of the TCP layer, which provides the reliable data flow service for the network application layer.

It mainly includes RM management sub-layer and RM reliable sub-layer, in which RM management sub-layer is responsible for establishing and releasing session connection and RM reliable sub-layer is responsible for the reliable transmission of communication data. The reliable active nodes in the network can only realize the reliable transmission of the data in the multicast communication layer and are not responsible for the network connection management. By adopting active network technology, the multicast communication layer contains a series of programs that can deal with the reliability of communication data. The model application layer realizes the dynamic configuration of active network multicast communication layer service by selecting reliable active nodes. According to the functional requirements of the model, reliable active nodes are divided into two types: precursors active nodes and reliable active nodes. The former is similar to the traditional Internet network node function, but it still needs to use the active network technology to realize the specific function. The latter can handle reliability work and requires the following functions:

（1）Active and reliable nodes can not only detect whether the communication data package is wrong, but also can timely process the communication data loss packet retransmission application of the lower node;

（2）Every active reliable node in the model guarantees its children to receive communication data correctly, which requires it to be able to realize communication data cache, so that the original data can be recovered in time when communication data is lost;

（3）The active reliable node can adopt local multicast to respond to the packet loss retransmission request of the active network sub node.

The hierarchical model of multimobile node routing communication protocol in active network is established.

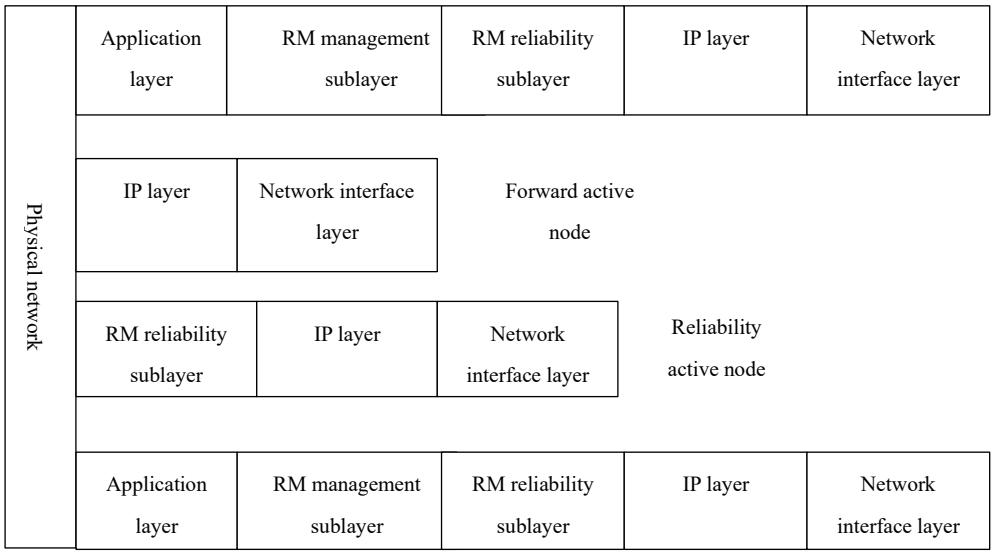


Figure 1 Hierarchical model of routing communication protocol for multiple mobile nodes

The efficiency of data packet transmission is improved.

2.2 High performance router multi mobile node communication protocol based on reliable active node

The multimobile node communication protocol of high performance router based on reliable active node mainly includes two parts: routing selection mechanism of active network hierarchical model and neighbor node management. The specific descriptions are as follows:

1）Routing mechanism of active network layered model

①Routing delay estimation

Suppose T_c is the time when the communication packet competes with the active network communication channel; T_t and T_p represent the transmission time and processing time of the communication packet respectively; T_q represents the queue delay time of the packet to be sent; T_s represents the sleep time of the active network node, and calculates the packet transmission delay time between the adjacent nodes of the active network.

$$Delay_{S\to i}(P_s) = T_c + T_t + T_p + T_q + T_s = \frac{T_2 - T_1}{2} \quad (10)$$

Among them, T_1 and T_2 represent the time when the communication packet recorded by the active network source node s enters the multicast communication layer and the time when the sub node receives the communication packet, respectively; $(T_1 - T_2)/2$ represents the single hop transmission delay time of active network communication packets. Based on the full consideration of communication energy consumption and transmission power, the weighted average method is used to update the packet transmission delay between adjacent nodes in active network in real time

The specific update formula is as follows:

$$Delay_{S\to i}^{i+1}(P_s) = aE(S,D)Delay_{S\to i}^{t}(P_s) + \frac{1-\alpha}{T}\sum_{k=\max(t,t-T)}^{i-1} Delay_{S\to i}^{k}(P_s) \quad (11)$$

Among them, T is the channel contention time window of active network; $Delay_{S\to i}^{t}(P_s)$ and $Delay_{S\to i}^{k}(P_s)$ represent the packet transmission delay time between adjacent nodes of active network at time t and time k respectively; $0 \le \alpha \le 1$, if the transmission delay of active network communication packet changes greatly, α takes the larger value; conversely, α takes the smaller value.

② Link quality of active network communication packet transmission

The quality of active network communication packet transmission link directly affects whether the packet can be reliably transmitted to the target node within a limited time; The routing mechanism of layered communication routing protocol based on reliable active node needs to consider the transmission link quality when selecting the next hop forwarding node of communication packet, so as to ensure the reliability and real-time of communication packet transmission. The communication link quality of active network is measured by the communication packet arrival success rate index. The specific measurement formula is as follows:

$$\mathrm{PRR} = \left[1 - (\frac{8}{15})(\frac{1}{16})\sum_{j=2}^{16}(-1)^j \exp\left(20\gamma(d)(\frac{1}{j} - 1)\right)\right]^{176} \quad (12)$$

Where $\gamma(d)$ is the signal-to-noise ratio of active network communication. j is the energy consumption value of network communication link, $i \in F(i)$.

③ Communication route selection

When the active network source node s needs to send packets, it needs to check whether the sink node is the next hop forwarding node.If so, the data packet can be sent directly from source node s to node sink without relay node; If not, it is necessary to calculate the forwarding adaptation index $\theta_{S\to i}$ of all nodes in the neighborhood node set F (i) that satisfy the following formula (13). Select the node with the largest value of θ s → I as the next hop forwarding node of the active network source node s, and calculate the forwarding adaptation index $\theta_{S\to i}$ according to the following formula (14).

$$v_{rep}(S,D) \le v_{S\to i}(P_s) \quad (13)$$

$$sink\theta_{S\to i} = \alpha \times \frac{v_{S\to i}(P_s)}{\sum_{i\in F(i)} v_{S\to i}(P_s)} + \beta\frac{E^{S\to i}(P_s)}{E_i} + \gamma \times PRR \quad (14)$$

$$\alpha + \beta + \gamma = 1 \quad (15)$$

Among them, α, β and γ are weight coefficients.

According to the above calculation, if all nodes in the set F(i) do not meet the formula (15), the following active network neighbor node management mechanism is started to search for the next hop forwarding node that meets the conditions.

2）Neighbor node management mechanism in active network

①In the active network, node i selects some nodes from the set F(i) to meet the requirements of packet transmission, and reduces the transmission power linearly;

②The neighbor node of active network finds that node i selects a part of nodes from the set F(i) and multiplies them with their respective packet transmission power to determine whether the selected nodes meet the requirements of transmission energy consumption and transmission power. If so, the value of $\theta_{S\to i}$ is calculated, and the node corresponding to the maximum value of $\theta_{S\to i}$ is selected as

the next hop forwarding node of packet transmission; If not, node i increases its own packet transmission power and updates the communication routing table until it finds the node that meets the transmission requirements.

3 Result analysis

During simulation,the active network is mainly composed of 100 nodes, and all nodes in the network are randomly distributed. Assuming that the initial energy of each node in the simulation test network is 0.5J, the network base station is selected according to the above-mentioned hierarchical model of active network communication routing protocol.

Compared with the traditional protocol, the change of the number of surviving nodes in the network with the simulation test time is shown in Figure 2, the number of surviving nodes of three different routing protocols gradually decreases. The change trend of the overall energy consumption of network nodes with the simulation test time is shown in Figure 3, the overall energy consumption of network nodes in routing protocol is low, which shows great energy saving advantages. The change of packet forwarding rate with the number of packet forwarding is shown in Figure 4. With the increasing number of packets to be forwarded in the active network, the packet forwarding rate of different routing protocols has decreased. The packet forwarding rate of the proposed routing protocol decreases the least because it selects the link with higher quality as the packet forwarding node, which effectively improves the forwarding rate.

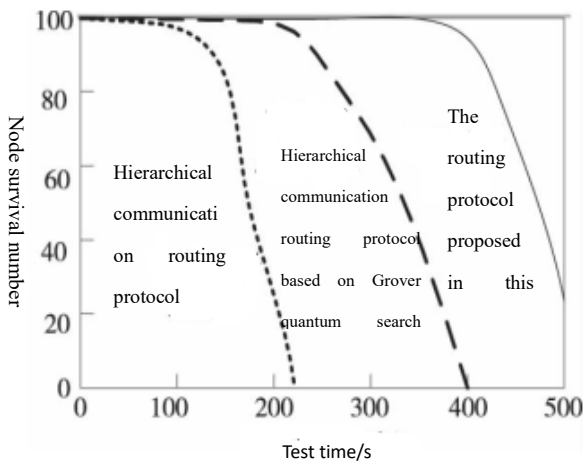


Figure 2 Comparison of the number of surviving

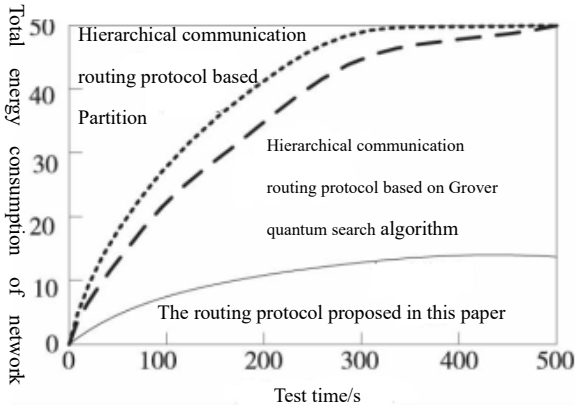


Figure 3 Total energy consumption of network nodes

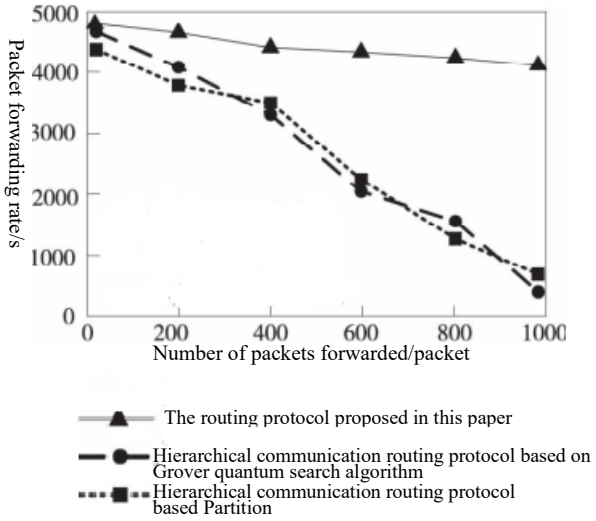


Figure 4 Comparison of packet forwarding rate

4 Conclusion

The high-performance router multi mobile node communication protocol based on reliable active nodes is proposed and designed in this paper. Through simulation and test, it is proved that the proposed routing protocol can effectively shorten the average end-to-end delay problem and improve the performance .

Reference

[1] Zhao canming, Li Zhuhong, Yan fan, etc. Routing protocol for load balancing in power communication networks . Computer applications, 2016, 36 (11): 3028-3032.

[2] Zhao Yue, Meng Bo, Chen Lei, etc.Energy aware routing protocol for wireless sensor networks. Computer engineering and design, 2016, 37 (1): 16-20.

[3] Fan Xincan, Liu Kaiyang, Wen Xiaojun. Research on transmission and routing protocol of quantum mobile Internet Communication. Journal of quantum electronics,

2017, 34 (5): 581-587.
[4] Zhai Chunjie, Xu Jianmin, Liu Yonggui. Energy Balanced Routing Protocol Based on partition . Journal of sensing technology, 2016, 29 (1): 80-87.
[5] Yao Yukun, Liu Jiangbing, Ren Zhi, et al. Efficient RPL routing protocol for centralized network congestion control . Systems engineering and electronic technology, 2017, 39 (12): 2810-2816.
[6] Yao Yukun, Zhang Qiang, Yang Jikai. High delivery rate opportunistic network routing protocol based on social group. Computer application research, 2017, 34 (2): 577-581.
[7] Lin Chunli, Cui Jie. Adaptive multipath secure routing protocol in WSN.Computation Mechanical engineering, 2016, 42 (6): 144-150.
[8] Wang Wei, Yi He. Security performance optimization design of wireless network optical fiber communication router. Computer simulation, 2017, 34 (9): 187-190.
[9] Huang Jinke, Fan Xiaoguang, Wan Ming, et al. Mobile ad hoc networks based on stable clustering Routing protocol. Journal of Beijing University of Aeronautics and Astronautics, 2016, 42 (11): 2332 -2339.

Research on MES System Based on Production Management of Railway Vehicle Reducer

Li Fuqiang
Qingdao Sifang Locomotive & Rolling Stock Co., Ltd.
Qingdao, China
e-mail: lifuqiang.sf@crrcgc.cc

Liu Jiaze
Zhengzhou University
Zhengzhou, China
e-mail: 214109285@qq.com

Zhang Jun
Zhengzhou University
Zhengzhou, China
e-mail: zhangjun@zzu.edu.cn

Zhang Xiangru*
Zhengzhou Machinery Research Institute Co., Ltd.
Zheng Zhou, China
e-mail: felan@126.com

***Abstract*—In recent years, the rapid development of rail transportation represented by high-speed railways and urban rail transportation has caused tremendous pressure on related discrete equipment manufacturing enterprises that are upstream of the locomotive and vehicle supply chain. The surge in orders has led to the company's "planning" and "production" It is difficult to achieve a consistent pace, production data cannot be uploaded and released, production and management are disconnected, and production efficiency cannot be improved. In the final analysis, there is a split between the underlying execution system and the enterprise management system in the workshop. In this regard, this topic studies and discusses the application of Manufacturing Execution System (MES) in discrete equipment manufacturing enterprises. In fact, it is aimed at the production and manufacturing enterprises of rolling stock reduction gears for rolling stock. The solution focuses on the process preparation subsystem based on the typical process library, and integrates it based on the MES platform. In the end, for enterprises to achieve the goals of reducing production preparation cycles, reducing inventory, and improving efficiency, help enterprises seize opportunities and meet challenges in the fierce market competition.**

Keywords- rail transit; reducer; MES; decision making.

I. INTRODUCTION

As a pivotal device for fast/slow switching, the locomotive reducer plays[1] a key role in regulating vehicle speed and maintaining smooth operation. In the high-speed rail speed and rapid development of rail transportation in the context of the rapid increase in demand for rolling stock, the reducer manufacturing enterprises are both opportunities and challenges. Reducer manufacturing is a typical discrete manufacturing, the distinctive feature of discrete manufacturing products by multiple parts through processing and assembly, each part needs to be processed through a number of discrete process processing, the production site is mostly workshop. The workshop is the main body of manufacturing, the use of information technology to upgrade the workshop production will help the locomotive reducer manufacturers seize the opportunity of development.

The most effective solution to these problems is the Manufacturing Execution System (MES), which was first proposed by Advanced Manufacturing Research (AMR)[2] and is defined as "a shop floor-oriented management information system located between the upper planning and management system and the lower industrial controls". At present, there is a lot of research on MES system and it has been applied in process manufacturing and discrete processing industry in many foreign countries to different degrees. For example, Consilium Corporation has developed MES I and MES II systems for the semiconductor and electronics industries. The U.S. Honeywell company developed POMSMES system for the pharmaceutical industry. Siemens Company of Germany has developed MES systems with industrial characteristics in aerospace, petrochemical, electric power, iron and steel and so on. Wang, Fa, Maw et al. focus on the application of MES systems in automotive manufacturing companies. Huibo Dong et al. study MES systems in the field of aircraft assembly. Bohui Ding and Jinxing et al. focus on MES system architecture in discrete manufacturing. Zhou Ke et al. focused on the design of MES systems for discrete manufacturing using RFID technology[3-6].

The traditional MES system[7-10] is based on a single software, which lacks interconnection with the actual environment and cannot transmit all kinds of real-time data in a timely manner, and the formulation of production plans and execution of production instructions are often not accurate, fast and comprehensive enough. At the same time, the system lacks self-adaptation and rapid response to unpredictable changes from the outside. In addition, it lacks the ability to quickly reconfigure for changes and adjustments to the manufacturing system. Another problem affecting the production efficiency of discrete manufacturing enterprises is the process design. Due to the production characteristics of the discrete manufacturing enterprises,

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00026

process processes are numerous and complex, associated elements, coupled with rolling stock reducer products with more varieties, batches, according to the characteristics of the order processing, making the process designers face tedious and repetitive work, affecting the product development cycle, which in turn affects the product production cycle.

This study is to explore the manufacturing execution system for the production of rail vehicle reducer of Zhengzhou Machinery Research Institute, which is suitable for the production of multiple varieties and small batches of rail vehicle reducer, and to integrate part of Computer Aided Process Planning (CAPP)[11] function with MES as the platform. Finally, the information integration of the production line can eliminate the communication barriers between production and management personnel, improve access to production data, workshop scheduling capabilities, quality control capabilities in the entire life cycle of the reducer production, and thus improve production efficiency, shorten the production cycle, reduce inventory, to help rolling stock reducer manufacturers seize the opportunity to meet the challenges.

II. System Requirements Analysis and Architexture Model

A. *Systen Requirements Analysis and Architecture model*

This section analyzes the requirements of MES system, which is divided into two aspects: system functional requirements and system non functional requirements.The success of the system largely depends on the quality of software system requirements analysis. In the actual system development process, due to the fuzzy, abstract, uncertainty, variability and other complex characteristics of requirements, and the professional barriers of reducer manufacturing make the requirements analysis of MES system becomes a complex task.In the process of MES system software requirements analysis, we consider the following points:

- The priority of requirements;
- The dependence of requirements;
- The complexity of requirements;
- The effectiveness of requirements.

The MES of railway vehicle reducer runs in the reducer production workshop to assist the production system to complete the whole process of transforming the input of production factors into the output of products. In order to realize the high-efficiency operation of the production system,there are three main functional requirements of process control,the management of production factors, the formulation and implementation of production plan, and the control of production process.

The factors of production in the enterprise manufacturing system include staff, machine, material, file and environment. The stuff is the core of enterprise manufacturing system, Other factors are all around the stuff, under the guidance of stuff arrangements for production.

The demand of MES system on production planning is mainly reflected in the following aspects,after the project is approved, the implementation plan is formulated according to the project content. According to the product BOM, the outsourcing production plan and the independent production plan are formulated. After the plan is formulated, the plan is distributed and fed back to the final warehousing.Production planning is the key to achieve the production objectives. Improving the accuracy of the plan is conducive to reducing the enterprise's inventory, or even achieving zero inventory. The system needs to have the ability to make and implement the production plan, which is embodied in the following aspects:

- ensure the smooth of production planning process;
- match the production plan and production capacity.
- Dynamic adjustment capability.

Production process control is one of the core functions of MES. Its requirements are as follows: monitor and control the production progress, assist the workshop director to allocate workshop production resources, and issue production tasks to workshop workers. It has three functions: production schedule monitoring, production planning and quality inspection.

System non functional requirements mainly include security requirements and ease of use requirements. The system should have high security requirements, be able to limit the login IP and login time period, improve the authority system, and assign different permissions to limit operations such as adding, deleting and modifying according to different security requirements, and record the user's data operation behavior in the form of log, so as to avoid the impact of human operation errors on system security.The system should be open to the administrator only due to the consideration of safety.

Most of the users of the system are the workers and managers working in the workshop. Due to the complexity of the site environment, it is necessary to make requirements for the usability of the software, try not to let the software operation become the burden of users, and improve the user experience as much as possible.

B. *Architecture Model*

According to the analysis of functional requirements and non functional requirements of the system, the overall functional model of MES can be preliminarily obtained. The overall architecture of the system is shown in Figure 1. The whole system consists of three modules: MES business function module, MES system integration interface and MES information display.

MES business function module is based on the MES integration model which contains 11 functions proposed by MESA. According to the actual situation and requirements of the enterprise, the business function module of MES is a customized business function model for the enterprise, which is composed of seven main function modules and several auxiliary functions. The main function modules are:

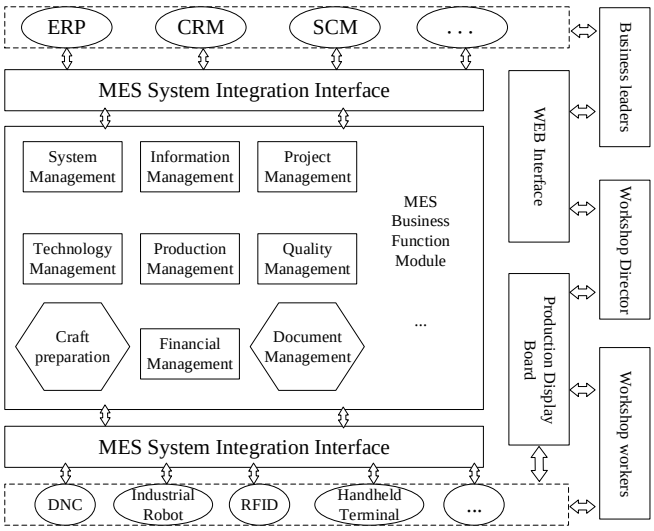


Figure 1. Overall system architecture

system management, information management, project management, technical management, production management, quality management and financial management module. Each function module contains multiple sub modules, and the sub modules contain multiple functions. The cooperation among various functions forms the main function group. The auxiliary function is independent of the function group and provides assistance for multiple function groups. MES business function layer is the core of manufacturing execution system. Through the interface layer of MES system, it connects with the enterprise management and the bottom execution system to complete the upload, release and processing of production information.

MES system integration interface is responsible for docking with upper management system and bottom execution system, which is the "central nerve" of manufacturing execution system.In connection with the upper management system, the design adopts the way of database integration, which can store the same data conditionally. If not, it provides a unified data interface module in the web server, and encapsulates the interface program in each system to realize the system integration.In connection with the bottom execution layer, because the underlying data has the characteristics of multi-source, heterogeneous and dynamic, it is necessary to standardize the field information data access, provide a unified way for MES business function modules to access the field data, and use OPC (OLE for process control) technology to solve the data access conflict, so as to provide the field data for the system.

MES information display provides data to enterprise leaders, workshop directors, workshop workers or other staffs in the form of web interface and production display board. By using the advantage of B/S architecture can break the geographical restrictions, users can access the Internet at any place and use the browser or mobile equipment to understand the real-time production situation.

III. MODEL DESINE OF MES SYSTEM

According to the business process analysis of the production workshop of railway vehicle reducer, the detailed requirement analysis of reducer MES system and the overall design of the system, the function points of MES software determined in this paper mainly include: system management module, information management module, project management module, technical management module, production management module, quality management module and financial management module.

A. System Management Module

The system management module is the basis of the whole business function, as shown in Figure 2, including staff management,department management,Workshop team management, role management, authority management, module management, a total of this six sub modules, the main function is to manage personnel information, authority system construction and function menu settings.The four sub modules of staff management, department management, workshop team management and role management are mainly used to manage the personnel information involved in the system. The three sub modules of role management, authority management and module management mainly complete the construction of authority system, and the module management mainly completes the configuration of function menu.

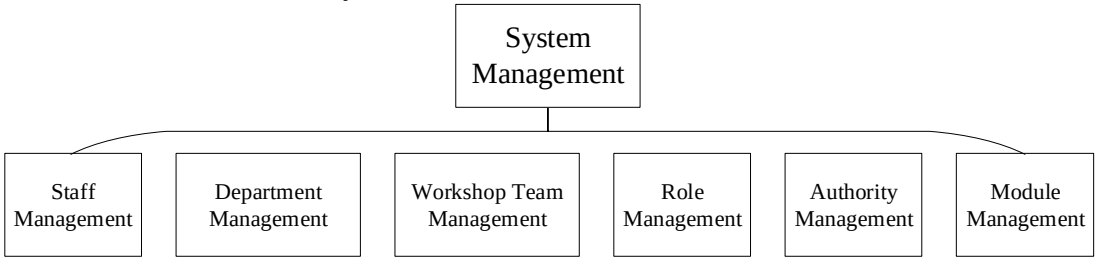


Figure 2. Structure of system management module

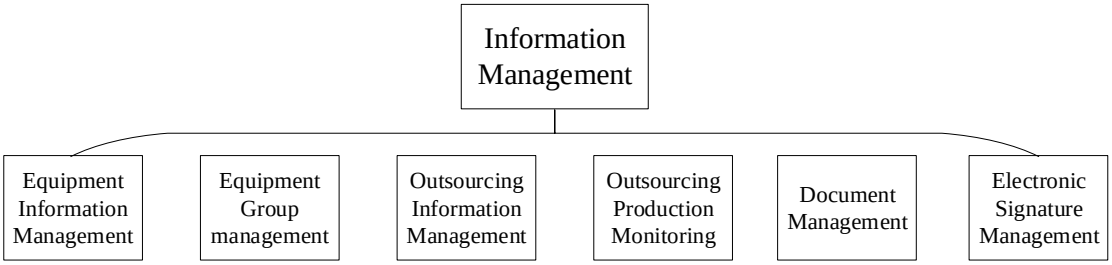


Figure 3. Structure of information management module

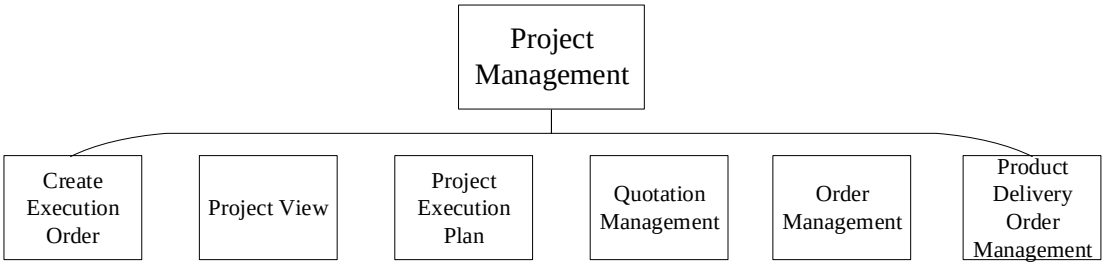


Figure 4. structure of project management module

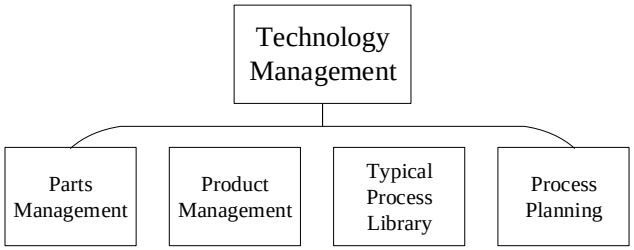


Figure 5. structure of technical management module

B. Maintaining the Integrity of the Specifications

The information management module is responsible for the management of other basic information except staff in the system, as shown in Figure 3, including equipment information management, equipment group management, outsourcing information management, outsourcing production monitoring, document management, electronic signature management. A total of six sub modules, the main functions of which are equipment management, outsourcing unit management, document management and electronic signature management .

C. Information management module

The project management module carries the enterprise production business process, including the whole process from project establishment to product delivery. As shown in Figure 4, it includes six sub modules: creation of execution order, project view, project execution plan, quotation management, order management and product delivery order management. These six sub modules work together to serve the enterprise production business process.

D. Technical management module

The technical management module is a module to manage the technical resources involved in reducer production, as shown in Figure 5, including parts management, product management, typical process library, process planning, a total of four sub modules.Among them, the part management sub module is the foundation of the product management sub module, because the product is composed of multiple parts.The product management sub module is responsible for coding the product features according to the product hierarchy based on the part library.Based on the concept of typical process knowledge, the typical process library is a module to manage the typical process of an enterprise. The sub module of process planning has two modes of manual and automatic, which can be used for process preparation.

E. Production management module

The production process is a complex process involving multi business, multi role and multi manufacturing resources. Production management can best reflect the characteristics of MES. The production management module is a module to manage the reducer production process.As shown in Figure 6, it includes production display board, production schedule,

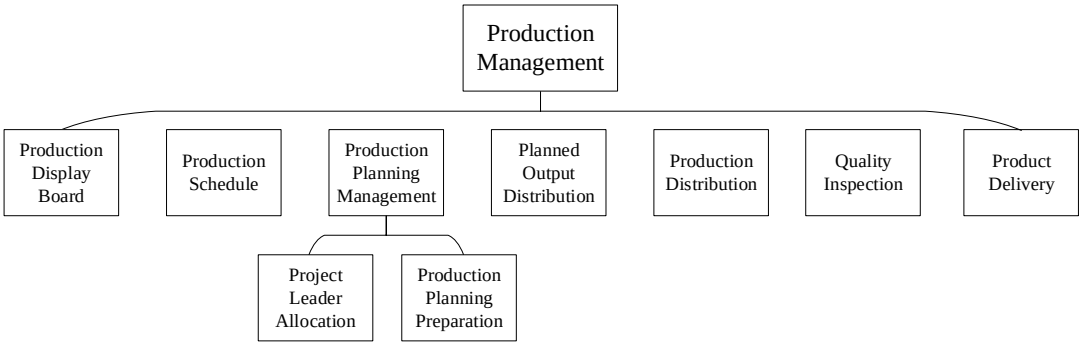


Figure 6. structure of production management module

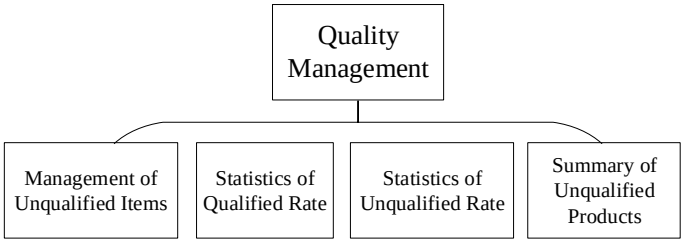


Figure 7. Structure of quality management module

production planning management, planned output distribution, production distribution, quality inspection, and product delivery, with a total of seven sub modules. Among them, the production plan management sub module can be divided into project leader allocation and production planning preparation.

F. Quality management module

With the development of the theory and practice of management science, quality management has become the most important part of enterprise management. It is equally important for MES. As a response to product quality, quality management can also reflect the implementation of MES based on workshop manufacturing execution system.The combination of quality management theory and MES technology can promote enterprises from quality inspection management stage and statistical quality management stage to total quality management stage.

The design idea of other sub modules of quality management module is: the quality inspection link completes the actual quality inspection process, which will produce a large number of data reflecting product quality. These data are the most valuable resources of the enterprise. Through these data, not only the quality data of products can be viewed, but also the production management level of enterprises can be reflected.This module mainly carries out the statistics and display of quality data, provides visual data for managers, and helps enterprises improve the level of quality management.As shown in Figure 7, it is mainly divided into four sub modules: management of unqualified items, statistics of qualified rate, statistics of unqualified rate and summary of unqualified products.

G. Financial management module

In the requirement analysis stage of reducer manufacturing execution system of the enterprise, through sorting out the requirements, it is found that part of the financial work is involved in the production process of the enterprise, and the system needs the support of the financial function to ensure the successful progress of the project and production activities.In addition, the purpose of workers' working hours statistics is to link the work of workers with their wages, and then to encourage workers' wages based on the principle of more work and more pay, which also needs to involve part of the financial functions. This module integrates some functions of financial management.As shown in Figure 8, there are four sub modules: payment management, invoice management, supplier debt management and working hours statistics. Invoice management can be divided into input invoice management and output invoice management.

IV. IMPLEMENTATION AND RESULT OF MES SYSTEM

A. System implementation

The MES system in this paper is customized and developed under the environment of Windows Server 2016.The system uses B/S architecture, visual studio 2017 as the development tool, and Microsoft SQL server as the data management system.The MES system realizes the mainstream functions of traditional MES, including basic system management, information management, project management, technical management, production management, quality inspection management and financial management.

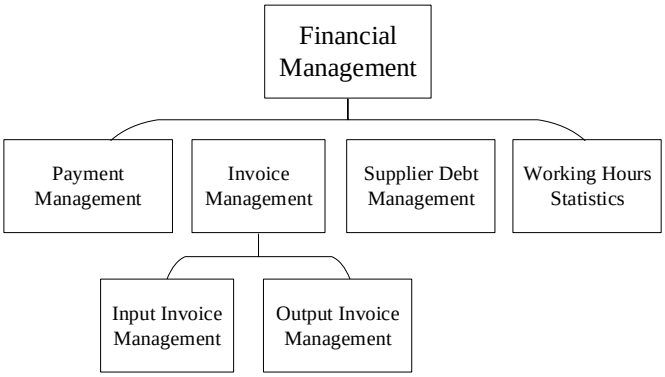


Figure 8. structure of financial management module

B. *Implementation result*

Finally, the system has been deployed in the enterprise and is in normal operation. The actual production process shows that the application of the system greatly improves the accuracy of production planning and the success rate of implementation, improves the efficiency of part process planning, and further improves the level of enterprise quality management. The effect of the system is summarized as follows:

- It optimizes the production business process of railway vehicle reducer, greatly improves the product quality of rail vehicle reducer, reduces some staff cost, and achieves a 25% year-on-year decrease in management cost.
- By changing the way of data collection, real-time data collection and delay reminder are realized in the production process. The efficiency of data collection is improved by 80%. The real-time traceability of product quality data is realized, which lays the foundation for product quality analysis.
- Forced to improve the management method, optimize the management process, through the substantial improvement of the implementation effect, the workshop optimizes part of the work process, makes the production management method closely combined with production, and forms a benign development.

V. CONCLUSIONS

This paper proposes a system solution for establishing MES for rolling stock reducers in rail transit to address the problems faced by rolling stock reducer manufacturers. We proposed a process preparation subsystem based on a typical process library, and integrated it with MES as a platform, completing requirements analysis, scheme design, functional design, development, and testing of the system. The successful implementation of the system improves the efficiency of production execution, shortens the processing coordination time and minimizes the maximum completion time.In addition, the workshop display board and schedule management module directly exposed the problems in the whole production process to the relevant managers, greatly improving the management efficiency.Realizing the steady progress of production planning and actual tasks, the gradual improvement of product quality, and the continuous accumulation of product data, which provides a strong reference for other reducer processing plants and discrete processing manufacturing industry.

REFERENCES

[1]. Zhao," Discussion on Design Method of Speed Reducer of Railway Locomotive," Mathematical Technology and Application , vol.25, pp.92-92 ,2010.

[2]. Pham, D.T., Eldukhri, E.E., "The Fp6 I*Proms network of excellence," Integrating European advanced manufacturing research. Korean Journal of Pain , vol.34, pp.166-171 ,2006.

[3]. Zhang G.J, D.Q., Wang LJ," Flexible workshop production scheduling optimization method," Computer Science, vol.18, pp.269-275 , 2018.

[4]. Yibo, D. "Research and development of MES system in CNC system production workshop," Manufacturing technology and machine tools, vol.16, pp.130-134 , 2016.

[5]. Ruchun, D., Kongkuai, G. "Research on Manufacturing Execution System MES Based on Lean Supply Chain," Industrial Engineering Management , vol.13, pp.114-120 ,2012.

[6]. Mingchuan, W., Pu, Y., Xiaoying, Y. "Research on Tractor Assembly Quality Management Information System Based on MES," Modern manufacturing engineering, vol.26, pp.51-58 ,2018.

[7]. Blanc, P., Demongodin, I., Castagna, P. "A holonic approach for manufacturing execution system design: An industrial application," Engineering Applications of Artificial Intelligence , vol.21, pp.315-330 ,2008.

[8]. Qingyun Daisupa/sup, R.Z.s.s. "Radio frequency identification-enabled real-time manufacturing execution system: A case study in an automotive part manufacturer," International Journal of Computer Integrated Manufacturing , vol.25, pp.51-65 ,2012.

[9]. Hwang, Y.D."The practices of integrating manufacturing execution system and six sigma methodology," International Journal of Advanced Manufacturing Technology , vol.30, pp.761-768 ,2006.

[10]. Molina, A., Santaella, A.R. "Achieving e-Manufacturing: multihead control and web technology for the implementation of a manufacturing execution system," Journal of Intelligent Manufacturing, vol.17, pp.715-724 ,2006.

[11]. Marri, H.B., Gunasekaran, A., Grieve, R.J. "Computer-aided process planning: A state of art," International Journal of Advanced Manufacturing Technology , vol.14, pp.261-268 ,1998.

A Survey of the Inadequacies in Traffic Sign Recognition Systems for Autonomous Vehicles

Angelica F. Magnussen
College of Engineering
University of Texas at Arlington
Arlington, TX, USA
angelica.magnussen@mavs.uta.edu

Nathan Le
College of Science and Engineering
Texas A&M University-Corpus Christi
Corpus Christi, TX, USA
nle10@islander.tamucc.edu

Linghuan Hu
Dept. of Computer Science
University of Texas at Dallas
Richardson, TX, USA
linghuan.hu@utdallas.edu

W. Eric Wong*
Dept. of Computer Science
University of Texas at Dallas
Richardson, TX, USA
ewong@utdallas.edu

Abstract—Traffic sign recognition systems are crucial for autonomous vehicles. They assist autonomous driving systems by collecting road-related information, such as speed limits, stop signs, etc., that are necessary for safe driving. However, as evidenced by recent autonomous vehicle crashes and recognition system failure-related studies, there are serious concerns about the inadequacies of the traffic sign recognition systems and their used techniques. In response to the industrial needs and to help practitioners improve the reliability and safety of the traffic sign recognition systems, this paper discusses the general architectural outline of traffic sign recognition systems and the challenges that must be overcome, in order for traffic sign recognition systems to be safe and reliable. An in-depth discussion of various solutions is given to provide practitioners valuable insight into the improvement of traffic sign recognition systems.

This paper has been recommended by the ISSSR 2020 Program Committee to the International Journal of Performability Engineering for possible publication. To avoid duplication, only the abstract of the paper is included in the Proceedings

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00027

A Local Feature Descriptor based on Improved Codebook Model

Wu Qinggang, Zhai Xueming
School of Computer and Communication Engineering
Zhengzhou University of Light Industry
Zhengzhou, China
e-mail: wuqinggang323@126.com

Yue Baohua
College of Information Engineering
Xinyang Agriculture and Forestry University
Xinyang, China
e-mail: 331617529@qq.com

***Abstract*—The commonly used local descriptors often suffer from the sensitivity to image rotation. To overcome such disadvantages, this paper proposes a new local feature descriptor based on an improved codebook of ascending permutation. The proposed descriptor is composed of four steps. Firstly, the original image is uniformly divided into numerous blocks. Then, the pixel values in each block are arranged into column vectors in clockwise. Subsequently, the pixel values in the column vector are sorted in ascending permutation. Finally, such pixel values that arranged in ascending order are used as the local feature descriptor for the given image block. The proposed local features can also be transferred to improve the classical local descriptors. To validate the effectiveness of the proposed descriptor, extensive experiments are conducted on Caltech 101 dataset, and the results demonstrate that the improved model of ascending permutation is more robust to image rotation than original ones.**

Keywords-code; codebook model; feature descriptor; image rotation; ascending permutation; image classification

I. INTRODUCTION

The local feature extraction and description play an important role in many computer vision tasks. The extracted local features represent essential properties of images and can help to correctly interpret the scenes [1]. To makes full use of the abundant information that implicitly embedded in images, the features of local regions should be properly extracted. The feature representation of the whole image is formed by combining the local image features, which can enhance the robustness of descriptors to various interference and obtain a relatively stable result. At present, local feature extraction has been widely used in image retrieval, image matching, object recognition, image segmentation and target localization, etc. [2].

The classical local feature extraction methods include SIFT, SURF, BRIEF, BRISK, FREAK and so on. The SIFT descriptor [3] constructs a 128-dimensional feature vector by grading the gray-scale gradient direction in a 4×4 sub-regions near the feature points. It is characterized by the fixed angle, scale, and brightness. However, when the viewpoint and illumination fluctuate to a great extent, the classification accuracy decreases dramatically. The SURF descriptor [4] uses the local response of Haar wavelet to establish a feature vector, which has good real-time performance. Similar to SIFT, the effectiveness of SURF will be invalid when the illuminance becomes blur. BRIEF [5] is a binary coded descriptor that describes the feature points. Although the calculation speed is fast, it is sensitive to noise and image rotation. The BRISK method [6] performs better for the case of blurry images, but the matching ability is poor when the scale changes [7]. The FREAK[8] algorithm exhibits strong ability to illumination changes. In recent years, the local feature extraction methods also include KAZE [9] and AKAZE [10]. The matching of KAZE on scale invariance is inferior to SIFT, but KAZE is more robust to image blurriness than other features under information loss, noise interference and compression reconstruction, etc.. The AKAZE is an accelerated version of KAZE. Compared with SIFT and SURF, AKAZE is faster, and the repeatability and robustness are also greatly improved compared with BRISK. However, all above local feature descriptors perform worse when image is rotated in a relatively large viewpoint. Although some of the descriptors have certain ability of robustness to image rotation, the computational burden is heavy.

To improve the ability of the commonly used feature descriptors to image rotation under the constraint of real-time application, a new local feature extraction algorithm is proposed in this paper based on an improved codebook model of ascending permutation. Firstly, the original image is divided into several image blocks, and then the pixel values in each image block are reshaped into column vectors. Subsequently, the pixel values in each column vector are re-arranged in ascending order, and such ascending ordered pixel values are taken as the local features of each image blocks.

The remainder of the paper is organized as follows. Section 2 introduces the related works of local feature descriptors including SIFT, SURF, BRIEF, and BRISK. Section 3 discusses the proposed local feature descriptor based on ascending permutation. In Section 4, extensive experimental results are presented. And finally, the conclusion is drawn in Section 5.

II. CLASSICAL LOCAL FEATURE DESCRIPTORS

There are numerous local features descriptors applied in image classification, image matching, or object recognition. In this section, we briefly introduce the principles of four feature descriptors (SIFT, SURF, BRIEF, and BRISK) since the proposed ascending permutation can also be used to improve the performance of such local feature descriptors.

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00028

A. SIFT

The SIFT descriptor [3] was firstly proposed by David Lowe in 1999. SIFT consists of the following four steps. Firstly, it constructs scale space, detects extreme points, obtains scale invariance, filters and locates feature points accurately. Then, it assigns direction values to the feature points, each of which has three information: location, scale and direction. Thus, a SIFT feature point can be determined. The feature point descriptor is constructed as shown in Figure 1. Since SIFT has many invariance advantages of affine, perspective, rotation and illumination, so it has been widely used in image feature extraction.

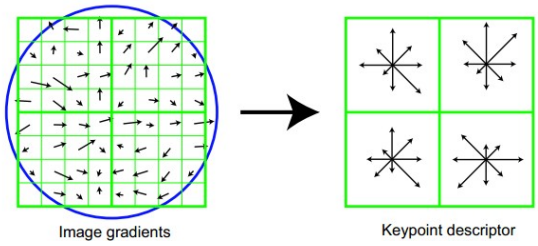


Figure 1. Construction Process of SIFT Descriptor.

B. SURF

SURF [4] was proposed by Herbert Bay etc. in 2006, which aims to deal with the shortcomings of SIFT (i.e., too much computation time). It uses the speckle feature detection based on Hessian determinant (DOH). Given an interested pixel in an image whose grayscale values can be represented as $I(x,y)$, the Hessian matrix with scale is defined as follows:

$$H(x,\sigma)=\begin{bmatrix} L_{xx}(x,\sigma)L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma)L_{yy}(x,\sigma) \end{bmatrix} \tag{1}$$

Where L_{xy} is the convolution between the second-order derivative of Gaussian filter $\frac{\partial^2}{\partial x^2}g(\sigma)$ and the given pixel $I(x,y)$, where $g(\sigma)=\frac{1}{2\pi\sigma^2}e^{-(x^2+y^2)/2\sigma^2}$. By constructing Hessian matrix, all interested points are generated. Then, the scale space is constructed to locate the feature points. At last, the principal direction of feature points is determined by the Harr wavelet eigenvalues in the circular neighborhood of these feature points. Accordingly, the descriptor of the feature points is obtained.

C. BRIEF

BRIEF [5] is a feature descriptor which need to determine the location of the feature points in advance. Then the image is smoothed with Gaussian filters. Meanwhile, taking the neighborhood window of S×S centered at feature points. Randomly selecting a pair of points in the neighborhood window, comparing the two pixel values, and the descriptor is defined as follows:

$$\tau(p;x,y):=\begin{cases}1, & if\ p(x)<p(y)\\ 0, & otherwise\end{cases} \tag{2}$$

where $P(x)$ and $P(y)$ are the gray values of two pixels $x=(u_1,v_1)$ and $y=(u_2,v_2)$, respectively. Then, N pairs of random points are randomly selected in the neighborhood window, and the binary values are assigned to form a binary code. The binary code describes the feature points, i.e. the feature descriptor. Generally, it is commonly set N=256. BRIEF abandons the traditional method of describing feature points using gray histogram in a given area, which greatly speeds up the process of constructing feature descriptors.

D. BRISK

To describe the feature point, BRISK [6] compares the gray values of pixels to get a cascade of binary bits. The uniform sampling mode is adopted when describing the feature points. In other words, the discrete Bresenham concentric circles with different radius are constructed which are centered at the feature points. Then, N sampling points with equal distance are obtained on each concentric circle, as shown in Figure 2 (scale = 1, N = 60). Since this neighborhood sampling may cause image aliasing, it is necessary to filter the sampling points on concentric circles with different radius. BRISK has neither strong rotation and scale invariances, nor robustness to variations.

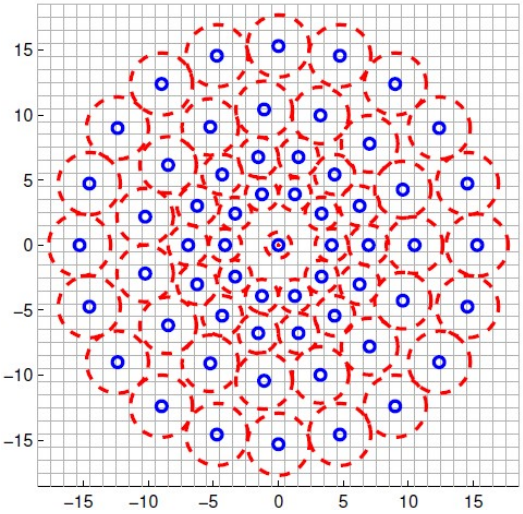


Figure 2. BRISK algorithm sampling mode.

III. CODEBOOK MODEL BASED ON ASCENDING PERMUTATION

In this paper, a local feature descriptor is proposed based on the ascending order of pixel values in a neighborhood, which is robust to image rotation. Let ϕ denotes the ascending order of pixel values, π represents the sort operation performed on the original feature vector.

$$\phi_0=(I_1,I_2,I_3,...,I_N) \tag{3}$$

$$\phi_\pi=\phi_0^{\pi}(I_{\min},I_2,I_3,...,I_{\max}) \tag{4}$$

To measure the similarities between two histograms, Histogram Intersection Kernel (HIK) [11] is exploited:

$$I\left(H\left(X\right),H\left(Y\right)\right)=\sum_{i=1}^{r}\min\left(H\left(X\right)_j,H\left(Y\right)_j\right) \quad (5)$$

where X and Y are given histograms. $H\left(X\right)_j$ is the j-th bin in X histogram, r is the number of histogram bins. The overlapping number of bins is the minimum of both different histograms. Thus, the overlapping number is defined as the HIK distance.

IV. Experimental Results and Analysis

A. Caltech101 Database

To evaluate the performance of the proposed local feature descriptor, the Caltech 101 image dataset [12] is adopted in the task of image classification in this paper. Caltech 101 is widely used to test the performance of various algorithms in the fields of computer vision and image processing. It is collected by F. F. Li etc. in 2003 and contains 9146 images which are divided into 101 categories (such as flowers, airplanes, faces, etc.). Each category contains different number of sample images from 31 to 800, and the size of which is about 300 × 200.

B. Experimental Settings

All the experiments are implemented on MATLAB R2014, and the system environment is Win 8, Inter (R) Core (TM) i5 CPU @3.20GHz,4.00GB RAM. As for the testing samples, the images belonging to both categories of Butterfly and Airplane in Caltech101 dataset are selected to evaluate the performance of the proposed algorithm. The Butterfly category contains 91 sample images, while the Airplane category consists of 800 sample images. In addition, there are two key parameters needed to be initialized in the proposed descriptor, i.e., block size and codebook size. Both parameters are set to 8 and 256 in all experiments, respectively. At last, HIK distance is exploited to measure the difference between two histograms in the experiments.

C. Experimental Results

To validate the effectiveness of the proposed local feature descriptor to image rotation, we randomly select a butterfly image and airplane image as shown in Figure 3 (a). Each image is rotated in three different angles, i.e., 90°, 180°, and 270°., and the results are shown in Figure 3 (b)-(d). To extract the local feature, the first step is to obtain 256 codebooks from each original image using the original permutation and ascending permutation. For the four images in each category including both original and rotated ones, the second step is to calculate the local features by exploiting the codebooks of original and ascending permutations as stated in the first step. The frequency of each codebook is calculated and the corresponding histograms of original and ascending permutations for the four rotated images of Butterfly No. 0015 and Airplane No. 0129 are shown in Figures 4 and 5, respectively. In these histograms, the horizontal axis denotes the codebook numbers and the vertical axis represents the frequency of each codebook. It can be observed that the frequencies of ascending permutation are significantly different from the original permutations.

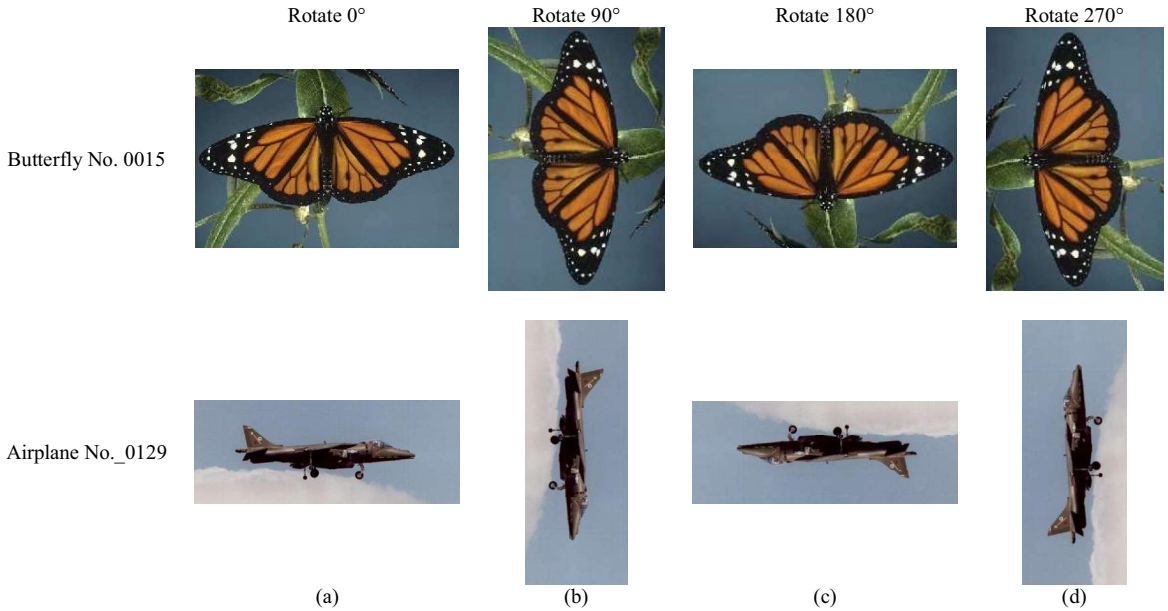
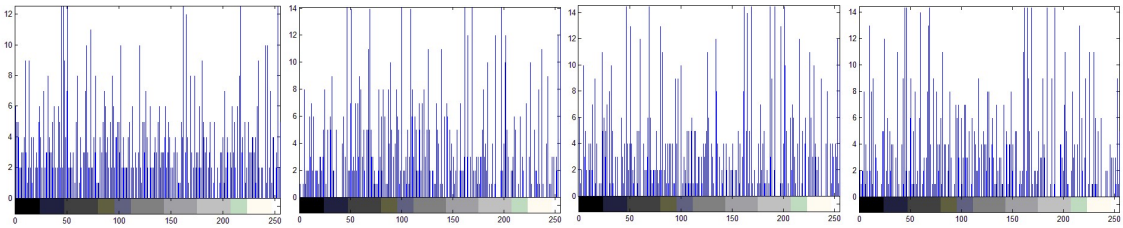
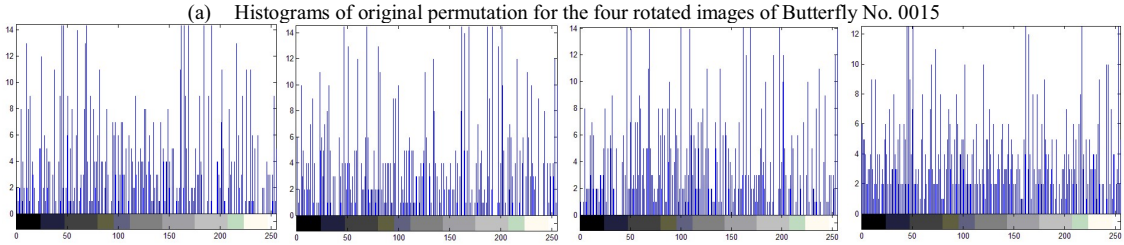


Figure 3. Rotated image of Butterfly No. 0015 and Airplane No. 0129.

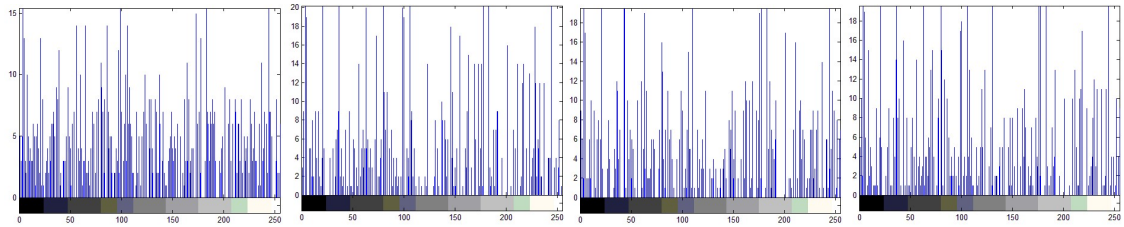


(a) Histograms of original permutation for the four rotated images of Butterfly No. 0015

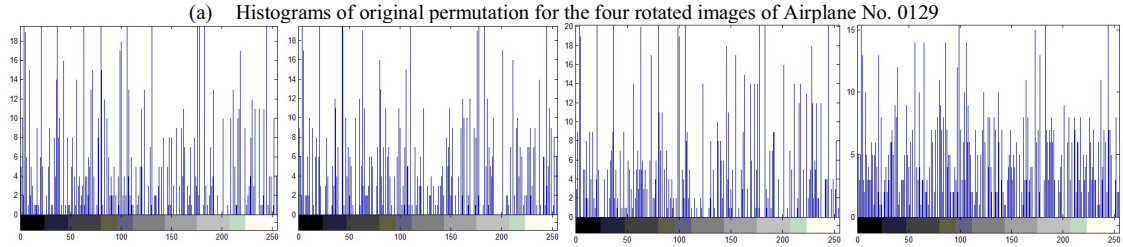


(b) Histograms of ascending permutation for the four rotated images of Butterfly No. 0015

Figure 4. Histograms of original and ascending permutations for the four rotated images of Butterfly No. 0015.



(a) Histograms of original permutation for the four rotated images of Airplane No. 0129



(b) Histograms of ascending permutation for the four rotated images of Airplane No. 0129

Figure 5. Histograms of original and ascending permutations for the four rotated images of Airplane No. 0129.

To quantitatively evaluate the performance of the proposed local feature descriptor, we calculate the HIK distances between the rotated images and the original ones (Butterfly No. 0015 and Airplane No. 0129) in terms of original and ascending permutations for above images, and the results are shown in Table 1. From this table, it can be observed that for both above images rotated by 90, 180 and 270 degrees, the HIK distances of ascending permutation are smaller than the original permutation. Note that the smaller the HIK distance is, the more similar between the rotated images and the original ones. Thus, the comparisons of HIK distances in Table I demonstrate that the codebook model of improved ascending permutation proposed in this paper is more robust to image rotation compared with the original codebook model.

TABLE I. COMPARISON OF HIK DISTANCE OBTAINED BY ORIGINAL AND ASCENDING PERMUTATIONS FOR BUTTERFLY NO. 0015 AND AIRPLANE NO. 0129

Image	Method	90° vs 0°	180°vs 0°	270°vs 0°
Butterfly No. 0015	Original	3.5773	4.0869	3.6336
	Ascending	2.9856	3.3296	2.8326
Airplane No. 0129	Original	5.1174	5.2878	5.1128
	Ascending	5.0296	3.1771	4.5826

To further validate the effectiveness of the proposed ascending descriptor, we calculate the HIK distances for all the 91 butterfly images in Caltech101 database in a similar way to above experiment. The HIK distances of images between the rotated ones and the original ones in terms of 90° vs 0°, 180° vs 0°and 270° vs 0°are shown in Figures 6, 7

and 8, respectively. The horizontal axis denotes the image number and the vertical axis represents the HIK distance. From these figures, it can be observed that most HIK distances obtained by the codebook model of ascending permutation (red line) is smaller than that of the original permutation (blue line) except for a few individual images. Although the results are not consistently satisfactory, the statistical results demonstrate that the ascending permutation is more robust than the original permutation.

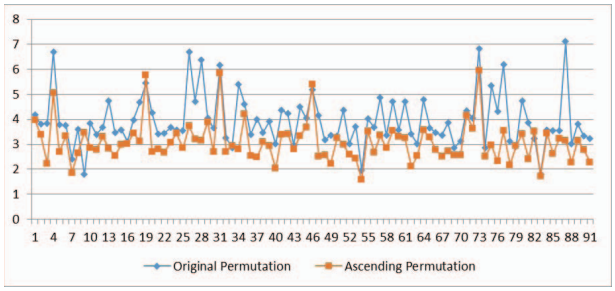


Figure 6. Comparisons of HIK distances between 91 rotated Butterfly images (90 degree) and original ones obtained by original permutation (blue line) and ascending permutation (red line).

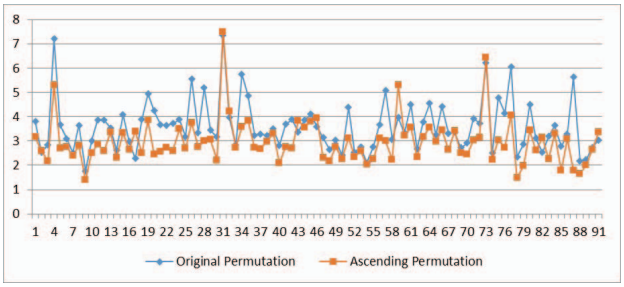


Figure 7. Comparisons of HIK distances between 91 rotated Butterfly images (180 degree) and original ones obtained by original permutation (blue line) and ascending permutation (red line).

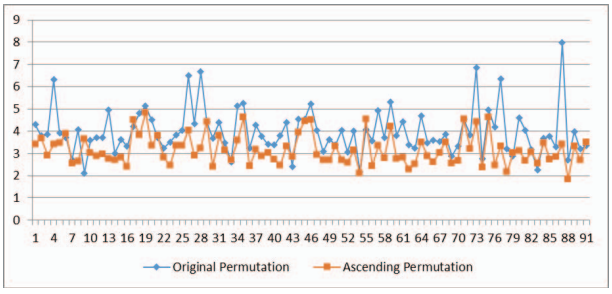


Figure 8. Comparisons of HIK distances between 91 rotated Butterfly images (270 degree) and original ones obtained by original permutation (blue line) and ascending permutation (red line).

In addition, we calculate the HIK distances for all the 800 airplane images in Caltech101 database in a similar way to above experiment. The HIK distances of images between the rotated ones and the original ones in terms of 90° vs 0°, 180° vs 0°and 270° vs 0°are shown in Figures 9, 10 and 11, respectively. From these figures, it can also be observed that most HIK distances obtained by the codebook model of ascending permutation (red line) is smaller than that of the original permutation (blue line) except for a few individual images, and the statistical results demonstrate that the ascending permutation is more robust than the original permutation.

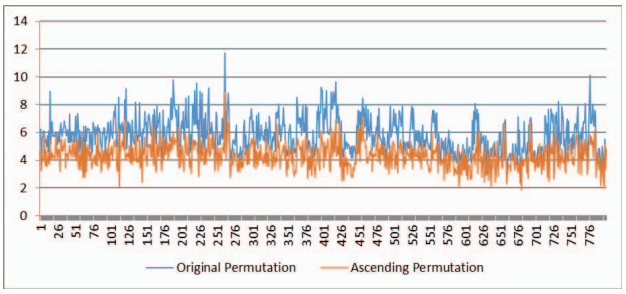


Figure 9. Comparisons of HIK distances between 800 rotated Airplane images (90 degree) and original ones obtained by original permutation (blue line) and ascending permutation (red line).

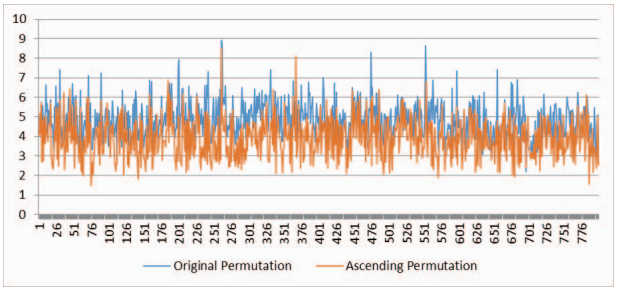


Figure 10. Comparisons of HIK distances between 800 rotated Airplane images (180 degree) and original ones obtained by original permutation (blue line) and ascending permutation (red line).

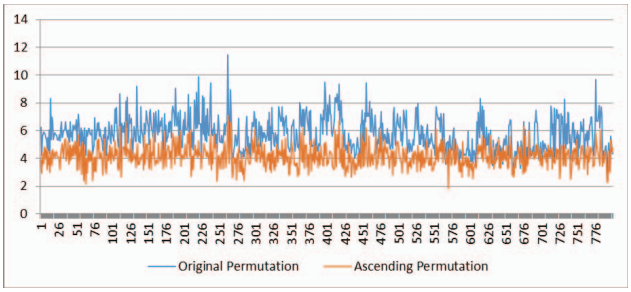


Figure 11. Comparisons of HIK distances between 800 rotated Airplane images (270 degree) and original ones obtained by original permutation (blue line) and ascending permutation (red line).

The average HIK distances on 91 butterfly images and 800 airplane images are summarized in Table 2. From this table, it can be observed that the average HIK distances of 91 Butterfly images are 3.8997 and 3.0480 by original and ascending permutation respectively for images rotated 90 degrees. The average distances are 3.5717 and 2.9413 for images rotated 180 degrees. The average distances are 3.9346 and 3.1347 for images rotated 270 degrees. So the proposed local feature descriptor is more robust in average. Similar conclusions can also be drawn from the average HIK

distances for 800 Airplane images as shown in the right three columns of Table II. In all, the proposed ascending permutation is more robust to image rotation compared with the original permutation.

TABLE II. THE AVERAGE HIK DISTANCE OBTAINED BY ORIGINAL AND ASCENDING PERMUTATIONS FOR 91 BUTTERFLY IMAGES AND 800 AIRPLANE IMAGES

	Butterfly images			Airplane images		
Mean values	90° vs 0°	90° vs 0°	90° vs 0°	90° vs 0°	90° vs 0°	90° vs 0°
Original	3.8997	3.5717	3.9346	5.8086	4.8550	5.7486
Ascending	3.0480	2.9413	3.1347	4.3352	3.9054	4.1562

V. CONCLUSION

In this paper, a local feature descriptor based on the improved codebook model of ascending permutation is proposed. Firstly, the original image is divided into several image blocks, and then the pixel values in image blocks are cascaded into column vectors. Then, the pixel values in column vectors are arranged in ascending permutation, and the pixel values in ascending order are taken as the local features of image blocks. Image classification results demonstrate that the proposed ascending descriptor is more robust to image rotation compared with original codebook model. More important, the proposed local features can also be used to improve the classical local descriptor. Compared with the traditional codebook model, the improved ascending codebook model is more robust to image rotation.

ACKNOWLEDGMENT

This work is supported in part by the National Natural Science Foundation of China under grant No. 61502435, in part by the Scientific and Technological Project of Henan Province under grant No. 14A520034, in part by the Doctorate Research Funding of Zhengzhou University of Light Industry under grant No. 2013BSJJ041, and in part by the young backbone teachers of Zhengzhou University of Light Industry under grant No. 2016042.

REFERENCES

[1] M. Yang, “Optical remote sensing image local feature extraction technology and its application research,” Master's degree, Cheng Du: University of Electronic Science and Technology Information, June, 2015.

[2] Biederman, “Recognition-by-components: A theory of human image understanding” Psychological Review, vol. 94, no. 2, pp. 115-147, April, 1987.

[3] D. G. Lowe, “Distinctive image features from scale-invariant key points,” International journal of computer vision (IJCV), vol. 60, no. 2, pp. 91-110, June, 2004.

[4] H. Bay, E. Andreas, T. Tinne, V. G. Luc, “Speeded-up robust features(SURF),” Computer Vision and Image Understand, vol. 110, no. 3, pp. 346-35, September, 2008.

[5] M. Calonder, V. Lepetit, C. Strecha, et al, “BRIEF: binary robust independent elementary features,” European Conference on Computer Vision. pp. 778-792, September, 2010.

[6] S. Leutenegger, M. Chli, R. Y. Siegwart, “RISK: Binary Robust invariant scalable key points” IEEE International Conference on Computer Vision, pp. 2548-2555, November, 2012 (DOI 10.1109/ICCV.2011.6126542).

[7] C. Suo, D. Yang, Y. Liu, ″Comparing SIFT、SURF、BRISK、ORB and FREAK in Some Different Perspectives,″ Beijing Surveying and Mapping, no. 4, pp. 23-26, 2014.

[8] R. Ortiz, “FREAK: Fast Retina Keypoint,” IEEE Computer Vision and Pattern Recognition, pp. 510-517, June, 2012.

[9] A. Bartoli , A. J. Davison, “KAZE features,” European Conference on Computer Vision. Springer-Verlag, pp.214-227, 2012.

[10] P. F. Alcantarilla, “Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces,” British Machine Vision Conference, 2013 (DOI 10.5244/C.27.13)

[11] P. Jia, N. Xu, Y. Zhang, “Automatic target recognition based on local feature extraction,” Optics and Precision Engineering, vol. 21, no. 7, pp. 1898-1905, July, 2013

[12] L. Fei-Fei, R. Fergus and P. Perona, “One-Shot learning of object categories,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 28, no. 4, pp. 594-611, April, 2008

A Survey on Automatic Bug Fixing

Heling Cao
Key Laboratory of Grain Information Processing and Control (Henan University of Technology), Ministry of Education
Zhengzhou, China
caohl@haut.edu.cn

YangXia Meng
College of Information Science and Engineering, Henan University of Technology
Zhengzhou, China
myx_ziqiboya@163.com

Jianshu Shi
College of Information Science and Engineering, Henan University of Technology
Zhengzhou, China
371329696@qq.com

Lei Li
College of Information Science and Engineering, Henan University of Technology
Zhengzhou, China
leili@haut.edu.cn

Tiaoli Liao
College of Information Science and Engineering, Henan University of Technology
Zhengzhou, China
tianli.liao@haut.edu.cn

Chenyang Zhao
College of Information Science and Engineering, Henan University of Technology
Zhengzhou, China
zhaochy2005@163.com

***Abstract*—To reduce the cost of software debugging, Automatic Bug Fixing (ABF) techniques have been proposed for efficiently fixing and maintaining software, aiming to rapidly correct bugs in software. In this paper, we conduct a survey, analysing the capabilities of existing ABF techniques based on the test case set. We organize knowledge in this area by surveying 133 high-quality papers from 1990 to June 2020 and supplement 57 latest high-quality papers from 2017 to June 2020. This paper shows that existing ABF approaches can be divided into three main strategies: search-based, semantic-based, and template-based. Search-based ABF considers using search strategies, such as genetic programming, context similarity, to change the programs into the correct one. Semantic-based ABF involves symbolic execution and constraint solving, such as satisfiability modulo theories solver, contracts, to fix bugs. Different from the two kinds of theories above, template-based ABF is mainly based on fixing templates, such as other programs, bug reports, to fix bugs. Besides, we provide a summary of the commonly used defect benchmarks and all the available tools that are frequently used in the field of ABF. We also discuss the empirical foundations and argumentation in the area and prospect the trend of future study.**

Keywords-automatic bug fixing; software debugging; search-based; semantic-based; emplate-based

I. INTRODUCTION

As a hot topic in the software maintenance field, ABF (Automatic Bug Fixing) attracts many researchers to study the repair models because of its ability to correct bugs and reduce software failures that cost millions of dollars [1]. In order to investigate and analyze the existing bug fixing approaches systematically, we filtrate the relevant papers that were cited in those papers we selected. Finally, we obtain total 133 high-quality papers published or accepted from 1990 to June 2020. Figure 1 summarizes the number of papers published per year from 2003 to June 2020. Overall, the number of papers is increased from 2003 to June 2020 and no relevant literature has been published only in 2004 and 2005. This trend indicates that this subject is receiving increasing attention. We also make a subdivision of journal papers and conference papers to calculate the percentage of conference papers published per year. According to our statistics, about 72% of the publications appeared in the conferences and workshops, it reflects the importance of the conference in this field.

At present, there exist several surveys on ABF [2-4]. Le Goues et al. [2] summarized the advantages and disadvantages of GenProg [5], a fitting example of how genetic search can be successfully adapted for fixing and discussed the current challenges faced by current research on ABF in 2013. Gazzola et al. [3] presented a conceptual framework that covers both healing and repairing solutions and illustrated the commonalities and differences between these two approaches in 2017. Monperrus [4] discussed behavioral repair and state repair in 2017, which is about automatically modifying the execution state at runtime.

Based on the summary of the above review, we further absorb the latest research results in this field. We supplement 57 latest high-quality papers from 2017 to June 2020 and illustrate the approaches of existing ABF approaches based on test case set. We also critically analyze the capabilities of each repairing method by comparing them with representative examples. Different from the work above, this paper shows that ABF approaches figure out the problem of repairing software according to three main strategies. Moreover, we provide a summary of the commonly used defect benchmarks and all the available tools that are frequently used in the field of ABF. In this paper, empirical foundations on fixing are detailed, including the argumentation in the research field. Besides, we summarize research challenges in ABF.

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00029

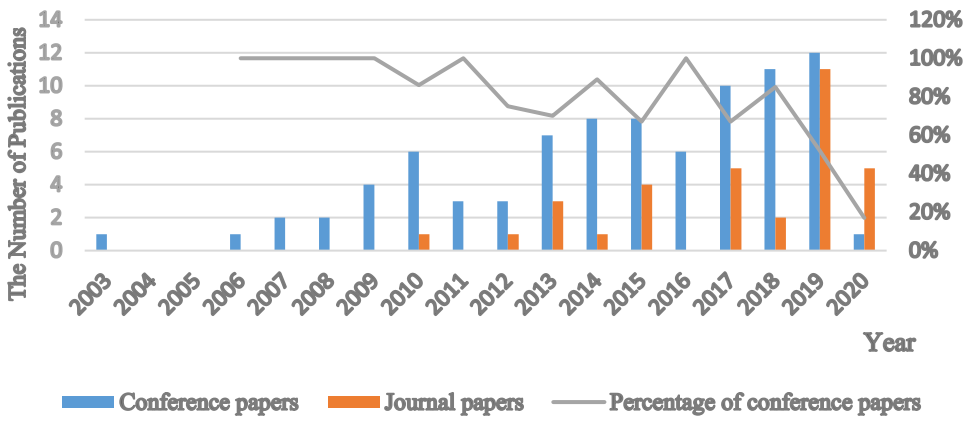


Figure 1. Illustration of literature by year.

The main contributions of this paper can be summarized as follows:

- We systematically analyze the research achievements made in recent years and collects 133 high-quality papers published by sorting, analysis, comparison, and summary.
- Bug fixing is classified into three categories according to the different fixing models: search-based bug fixing, semantic-based bug fixing, and template-based bug fixing.
- We provide a summary of the commonly used defect benchmarks and all the available tools that are frequently used in the field of ABF.
- We also summarize the empirical foundations and argumentation in the research field for the past few years.
- We finally discuss the current challenges which can influence future research in this area.

The rest of this paper is organized as follows. Section II presents three categories of bug fixing methods. Section III summaries the commonly used defect benchmarks and collects all the available tools. Section IV reviews the empirical foundations and argumentation in the research field. Section V discusses the existing challenges. Section VI provides some conclusions and future work.

II. ABF APPROACHES

How do developers fix bugs automatically? The following work [6, 7-10] answered this question. In terms of these different fixing models, ABF methods can be divided into three categories: (1) search-based bug fixing; (2) semantic-based bug fixing; (3) temple-based bug fixing.

A. Search-Based Bug Fixing

Search-based techniques intend to search a space of candidate solutions for solving a specific problem. Researchers have concentrated on the application of search techniques to repair bugs automatically. The search-based techniques mainly attempt to use existing code to replace the defective code for fixing and has achieved certain results in single bug fixing. However, there exist many problems, such as large search space, low search efficiency, and so on, in the current techniques based on search. The readability and accuracy of generation patches also need to be further improved. Besides, when there are multiple bugs in the program, the search space will grow exponentially. Therefore, this kind of technique performs poorly in the problem of multiple defects repair, which needs to be further explored. Currently, search-based methods focus on search strategies such as simulated annealing, genetic algorithm, ant colony algorithm, particle swarm optimization, and context similarity [5, 11, 12, 13].

Genetic programming was firstly adopted by Arcuri et al. [7] to evolve programs that met a formal specification for generating an oracle. Based on their previous work, Arcuri and Yao [14] presented a competitive co-evolutionary approach which used genetic programming and search-based software testing. Afterward, Arcuri [15] presented an evolutionary approach to automatically fix bugs by evolving the programs with a fitness function. The premise of the approach was that either a formal specification or a series of unit tests could be obtained.

To fix the real bugs, Weimer et al. [16] presented an automatic patch generation method which randomly mutated the buggy program. Furthermore, Weimer et al. [44] presented an approach to repair bugs combining program analysis techniques with evolutionary computation (i.e., mutation and crossover) and evaluated the effectiveness of repair bugs in off-the-shelf C programs. This approach took an abstract syntax tree (AST) as input and did not rely on formal specifications, which made it suitable for a wide range of software. Moreover, research by Wong et al. [17, 18, 19, 20] proposes a strategy for automatically fixing faults in a program by combining the ideas of mutation and fault localization. Based on the above consideration, GenProg [5] evolved the program variant at the statement level of AST that kept necessary functionality whereas avoiding a bug. Compared to GenProg, RSRepair [11] replaced the genetic algorithm with a random search for the problem that the genetic programming algorithm in the GenProg could not be useful for generating patches.

In order to minimize the number of candidate patches generated, Weimer et al. [21] presented a method AE which was based on the close semantic equivalence relation to

identifying candidate patches of semantic equivalence. Fast et al. [22] enhanced evolutionary method by designing a better fitness function with an extensive test suite set and dynamic predicates. In a subsequent study, the authors [23] extended the previous work [5, 7, 8, 16, 24] in several aspects, and the experimental results indicated that their approach could repair 55 out of 105 bugs, and outperformed the previous work [9, 10].

Tan et al. [25] recommended using anti-patterns to disable a series of transformations, because that the modified template would limit the search space and make the repair more inclined to use the pre-provided repair template. Wen et al. [12] proposed a context-based patch generation technique CapGen. CapGen worked at a fine-grained AST node level, considering AST node types and suspicious code elements of the required components when selecting mutation operators. 15 of the 21 bugs repaired by CapGen had never been repaired by existing approaches [26]. Six of the 21 bugs repaired by CapGen could be repaired by HDRepair [13]. Comparing with ACS [27], CapGen could repair 19 different bugs. ACS only targeted at synthesizing predicate conditions, and thus other bugs, in general, were out of the repairing capabilities. Simfix [28] combined two data sources, the existing patches, and the source program. It searched for correct patches within the intersection of two search space. In subsequent research, Jiang et al. [29] manually analyzed 50 real-world defects from Defects4J.

DeepRepair [30] prioritized and transformed the repair ingredients by using code similarities which were based on deep learning. DeepRepair is an early attempt to integrate machine learning in a program repair loop. DeepRepair leverages learned code similarities, captured with recursive autoencoders, to select repair ingredients from code fragments that are similar to the buggy code. DeepRepair uses machine learning to select interesting code, while SEQUENCER[31] uses machine learning to generate the actual patch. SEQUENCER generates patches which pass the test suite for 19 bugs and patches which are semantically equivalent to the human-generated patch for 14 bugs.

Liu et al. [32] presented LSRepair to search for repair ingredients at the method-level granularity by using three strategies of similar code search. They focused on method-level repair ingredients to limit the explosion. LSRepair could correctly fix 19 bugs from Defects4J, and other ABF tools did not yet repair 10 of them.Koyuncu et al.[33] integrated the mined patterns to an automated program repair tool, FixMiner, with which can fix 26 bugs of the Defects4J benchmark.

B. Semantic-Based Bug Fixing

Semantic-based techniques directly utilize semantic information to synthesize patches by symbol execution and constraint solving. It takes the repair constraint as the protocol synthesized by the program and generates patches with the constraint solver. The semantic-based bug fixing synthesizes code according to the constraints that a program must satisfy, and do not rely on “redundancy assumption” [34]. At present, semantic-based techniques [35, 36, 10], [37, 38] have tried to fix bugs in assignment statements, predicates, multiple bugs.

AutoFix-E [10] attempted to bridge the gap between specification-based and test-based fixing. Boolean queries were utilized to construct an abstract notion of the state to denote contracts of the class. A subsequent program synthesis effort could provide fixed statements. This technique could only correct violations of simple assertions. Demsky et al. [39] utilized the invariant detection tool Daikon [40] to generate candidate consistency properties of the data structure for data structure repair. Gopinath et al. [37] presented a semi-automatic program repair method based on specification and constraint solving.

SemFix [36] combined symbolic execution and constraint solving with program synthesis. AutoFix-E2 [41] provides a repair approach with contracts such as preconditions, postconditions, and class invariants. To automatically repair broken unit tests, ReAssert [42] used symbolic execution to address the program deficiencies. Malik et al. [43] studied a case of data structure repair based on the constraint. VEJOVIS [44] combined static analysis with dynamic analysis to localize the code lines in the backward slice of the related parameters and assignments. It utilized a string solver to identify candidate replacements of elements and generated the candidate values in the backward slice for fixing bugs automatically.

Samimi et al. [38] presented two ABF methods implemented as plugins of Eclipse for HTML bugs. The first method, QuickFix, fixed simple bugs by statically analyzing a constant print statement; whereas the second one, PHPRepair, dealt with more general repairs by dynamically analyzing the interactions among multiple statements. From the perspective of the simplified patch, Mechtaev et al. [45] attempted to integrate software defect location and program patch generation into one step and proposed DirectFix method. It converted ABF into the maximum satisfiability problem and generated patches with component-based program synthesis technology [46].

Xuan et al. [47, 6] focused on the repair of conditional statement defects and presented the Nopol method. It mainly fixed defects by modifying existing if conditional statements or adding preconditions before statements. Durieux et al. [48] designed a new code composition engine, DynaMoth, which based on dynamic exploration for Nopol [47] methods. Method calls could be included in the if condition patch generated by the DynaMoth engine, further widening the search scope for finding the correct patch.Ke et al. [49] presented the SearchRepair method. They believed that semantic code search [50] could be used to find code segments that were semantically like defective code from many open source project hosting websites. The results of empirical research based on IntroClass defect benchmark [51] showed that 150 defects could be repaired by SearchRepair, among which 20 defects could not be repaired by GenProg [5], RSRepair [11] and AE [21].

Xiong et al. [27] studied refined ranking techniques for condition synthesis and presented a tool, ACS, which achieved a relatively high precision (78.3%) and a reasonable recall (8.0%) on Defects4J benchmarks. ACS implemented as a Java program repair tool based on the source code of Nopol [47] and the bug localization library GZoltar [52]. Bhatia et al.

proposed Neuro-symbolic [53] which combined neural networks with constraint-based reasoning. For a given programming assignment, RNN was trained by using a corpus of correct submissions. JFix [54] was a semantics-based repair framework for Java programs that implemented atop Symbolic PathFinder [55]. The experiment showed that JFix could repair a variety of classes of bugs in large real-world Java programs and multi-line fixes.

C. Template-Based Bug Fixing

Template-based bug fixing approach is mainly based on fixing templates to repair defects. The templates are extracted, either manually or automatically, from historical data. Those approaches fix bugs according to a set of change operators. The operators are extracted from a sample set of patches that have been already used to repair. Template-based techniques [56, 57, 58] can generate more acceptable patches than search-based techniques and semantic-based techniques. According to the templates, researchers define a set of atomic operators to guide the generation of candidate patches. The patches are generated from the existing correct programs, so they are more readable for developers.

Gao et al. [57] utilized question-and-answer websites to fix defects. Tan et al. mainly studied on the ABF of regression defects and proposed relifix [58]. They manually analyzed 73 actual software regression defects from the CoREBebch library [59] and extracted 14 kinds of code conversion operations. Sidiroglou-douskos et al. [60] presented CodePhage which could handle many particular defects, including integer overflow, buffer overflow, and so on.

R2Fix [61] generated candidate patches from bug reports filed by users and used a range of pre-defined fixing templates. The identified bug reports and the corresponding templates were applied systematically to the buggy program, and the candidate solutions were validated running the test suite attached to the bug report. Similarly, reference [62] uses bug reports to fix bugs. Kim et al. [63] analyzed 62,656 manually written patches from Eclipse JDT, an open source project, and summarized ten commonly used code modification templates. Based on the above code, they presented the PAR approach, which had been empirically showed that it could successfully fix 27 of 119 actual Java defects.

Instead of directly generating program patches, Minthint [64] tried to generate modification tips to assist developers in manual generating patches. In order to verify the repairability of NPEfix[65], the authors carried out two sets of experiments, respectively targeting 11 real defects and 519 defects implanted in three open source programs. The experimental results showed that NPEfix could successfully repair 10 of 11 real defects and 318 of 519 implant defects.

Gupta et al. [66] proposed to repair common programming errors by deep learning and presented an end-to-end tool, DeepFix, which could fix multiple bugs in a program without relying on any external tool. DeepFix was a multi-layer sequence-to-sequence neural network with attention, containing an encoder RNN to process the input and a decoder RNN with attention to generate the output. The network was trained to predict the location of the bug and the correct statement. DeepFix can fix 1881 bugs completely and 1338 bugs partially from the 6971 erroneous C programs.

Le et al. [13] focused on analyzing the similarity between candidate patches and patch history libraries when evaluating patch selection. In their empirical study, this approach could successfully repair 23 defects, whereas GenProg approach [5] and PAR approach [63] could only successfully fix one defect and four defects, respectively. Moreover, Le et al. [67] presented a scalable repair synthesis engine, called S3, which used programming-by-examples method to synthesize high-quality patches for bugs.

TABLE I. DEFECT BENCHMARKS

Defect Benchmark	Programing Language	Total Number of Defects	Year of Publication	Related Literature	Download Address
Simens	C	90	2013	[68]	https://sir.csc.ncsu.edu/portal/index.php
ManyBugs	C	185	2009	[51]	https://repairbenchmarks.cs.umass.edu/
IntroClass	C	778	2015	[51]	https://repairbenchmarks.cs.umass.edu/
Defects4J	Java	438	2014	[69]	https://github.com/rjust/defects4j
Codeflaws	C	3902	2017	[70]	https://codeflaws.github.io/
DBGBench	C	27	2017	[71, 72]	https://dbgbench.github.io/
QuixBugs	Multilingual	40	2017	[73，74]	https://jkoppel.github.io/QuixBugs/
DroixBench	Android	24	2018	[75]	https://github.com/stan6/droixbench
Bugs. jar	Java	1158	2018	[76]	https://github.com/bugs-dot-jar/bugs-dot-jar
BEARS	Java	251	2019	[77]	https://github.com/bears-bugs/bears-benchmark
BugSwarm	Multilingual	3091	2019	[78,79,80]	http://bugswarm.org

III. BENCHMARKS AND TOOLS

A. Defect Benchmarks

High-quality defect benchmarks are indispensable to evaluate the effectiveness of ABF approaches. High-quality defect benchmarks need to meet the following requirements: (1) defects need to come from actual projects; (2) the scale of evaluation program is sizeable; (3) test case set can better cover the expected behavior of the program with high code coverage and correct test oracle. This paper summarizes the commonly used and shared defect benchmarks in current researches shown in Table 1.

Le Goues et al. [51] presented two defect benchmarks ManyBugs and IntroClass based on the C programming language. These defect benchmarks were used to evaluate the

effectiveness of ABF approaches. ManyBugs collected 185 bugs from nine sizeable open source projects (fbc, gmp, gzip, libtiff, lighttpd, php, python, walgrind, wireshark). It mined defects by analyzing version control systems which contained 5.9 million lines of code and more than 10,000 test cases. IntroClass came from the six C code problems in an undergraduate course, which contained 956 defects submitted by 200 students. These defects were divided into two groups: the first group contained 762 defects which could not be wholly passed in the matching black box test case set; the second group contained 810 defects that did not pass the set of white box test cases automatically generated with the KLEE tool [81].

Simens assemblies in SIR [68] contained different types of applications, each of which was matched with many test cases and multiple defect versions. Each defect version was generated by artificial defect injection. Defects4J [69] had extracted 438 defects and passed strict peer review by mining six Java projects (Commons Lang, JFreeChart, Commons Math, Joda-Time, Mockito, and Closure Compiler). Tan et al. [70] presented the Codeflaws benchmark consisting of 7436 programs in the Codeforces online database. Each programming contest consisted of multiple programming problems with various difficulty levels. Each program represented one user submission for a specific problem to Codeforces.

Böhme et al. [71, 72] fixed 27 real bugs from CoREBench which were extracted systematically from the 10,000 most recent commits and the associated bug reports. Then they compiled their study data for all 27 bugs into DBGBench to facilitate the practical evaluation of automatic bug localization, diagnosis, and repair techniques. QuixBugs benchmark [73] investigated cross-language performance by multi-lingual program repair tools. It consisted of 40 programs from the Quixey Challenge translated into Python and Java, each with a bug on a single line. It was the first multi-lingual parallel corpus of program repair benchmarks.

Tan et al. prepared a benchmark DroixBench, a set of 24 reproducible crashes in 15 open source Android apps, to verify the repair efficiency of their methods. Bugs.jar [76] was a large-scale dataset for research in automated debugging, patching, and testing of Java programs. It contained 1,158 bugs and patches from 8 sizeable open source Java projects. Madeiral et al. [77] presented a benchmark BEARS which contained 251 reproducible bugs from 72 projects by using the Travis CI and Maven build environment. Different from Defects4J and Bugs.jar, BEARS used CI (builds) to identify buggy and patched program version candidates. The BUGSWARM[78] toolkit has already gathered 3,091 fail-pass pairs, in Java and Python, all packaged within fully reproducible containers. Furthermore, the toolkit can be run periodically to detect fail-pass activities, thus growing the dataset continually.

B. Tools

In order to better support the recurrence of empirical studies, many researchers have shared automatic bug fixing tools. This paper summarizes the commonly used and shared defect benchmarks in current researches shown in Table 2.

Matias et al. [82, 83] presented Astor, which was a framework developed in Java and encoded the design space of generate-and-validate program repair approaches. The framework contained the implementation of 6 repair approaches, jGenProg, jKali, jMutRepair, which could be used in comparative evaluations. Astor can repair 98 real bugs from the Defects4J dataset. It also provided twelve extension points, which researchers could either reuse or extend for further research. Repairnator [84] was an autonomous agent that regularly monitored test failures, reproduced bugs, and ran program repair tools against each reproduced bug. It used three different program repair systems, had studied 11523 test failures over 1609 open-source software projects hosted on GitHub and generated patches for 15 different bugs.

TABLE II. ABF TOOLS

Bug Fixing Tool	Programing Language	Year of publication	Related Literature	Approach	Download Address
GenProg	C	2012	[5]	Search-Based	https://github.com/squareslab/genprog-code
Angelix	C	2016	[85]	Semantic-Based	https://github.com/mechtaev/angelix
DeepFix	C	2017	[66]	Template-Based	https://bitbucket.org/iiscseal/deepfix
MintHint	C	2014	[64]	Semantic-Based	https://bitbucket.org/iiscseal/minthint
RSRepair	C	2014	[11]	Search-Based	http://qiyuhua.github.io/projects/rsrepair/
SearchRepair	C	2015	[49]	Semantic-Based	https://github.com/ProgramRepair/SearchRepair
SemFix	C	2013	[36]	Semantic-Based	https://github.com/mechtaev/angelix
ACS	Java	2017	[27]	Semantic-Based	https://github.com/Adobee/ACS
HistoricalFix	Java	2016	[13]	Template-Based	https://github.com/xuanbachle/bugfixes
JAID	Java	2017	[86]	Template-Based	https://bitbucket.org/maxpei/jaid/wiki/Home
NPEFix	Java	2015	[65, 87]	Template-Based	https://github.com/Spirals-Team/npefix
Nopol	Java	2017	[47, 6]	Semantic-Based	https://github.com/SpoonLabs/nopol/
JFix	Java	2017	[54]	Semantic-Based	https://xuanbachle.github.io/semanticsrepair/
Repairnator	Java	2018	[84]	Mixing	https://github.com/Spirals-Team/repairnator
SimFix	Java	2018	[28]	Search-Based	https://github.com/xgdsmileboy/SimFix
CapGen	Java	2018	[12]	Search-Based	https://github.com/justinwm/CapGen
LSRepair	Java	2018	[32]	Search-Based	https://github.com/AutoProRepair/LSRepair

Astor	Java	2019	[82,83]	Mixing	https://github.com/SpoonLabs/astor
AVATAR	Java	2019	[88]	Template-Based	https://github.com/SerVal-DTF/AVATAR
TBar	Java	2019	[89]	Template-Based	https://github.com/SerVal-DTF/TBar
Maple	C	2019	[90]	Semantic-Based	https://maple-repair.github.io
ssFix	Java	2017	[91,92]	Search-Based	https://github.com/qixin5/ssFix
Sharpfix	Java	2019	[93,92]	Search-Based	https://github.com/sharpFix18/sharpFix/tree/master/expt0
SOSRepair	C	2019	[94]	Semantic-Based	https://github.com/squaresLab/SOSRepair
PraPR	Java	2019	[95]	Template-Based	https://github.com/prapr/prapr
SEQUENCER	Java	2019	[96]	deep learning	https:// github.com/kth/SequenceR
MPPEngine	Java	2020	[97]	Search-Based	https://github.com/yazhiniv/astor/tree/MPPEngine
FixMiner	Java	2020	[98]	Search-Based	https://github.com/SerVal-DTF/fixminer_source

Liu et al. built TBar, a direct APR tool, which systematically tries to apply fix patterns to program bugs. A thorough evaluation of the TBar was conducted on the Defects4J benchmark, and experimental results show that 43 Bugs in Defects4J are properly repaired. Based on ssFix[91], Qi Xin and Steven P. Reiss developed a new repair technique sharpFix[93],which follows ssFix's basic idea but differs significantly in the approaches used for code search and code reuse.

Afzal et al. create SOSRepair[94], an automated program repair technique that uses semantic code search to replace candidate buggy code regions with behaviorally-similar (but not identical) code written by humans. On a subset of the ManyBugs benchmark of such defects, SOSRepair produces patches for 22 (34%) of the 65 defects, including 3, 5, and 6 defects for which previous state-of-the-art techniques Angelix and GenProg do not, respectively. Unlike previous experiments, in the paper by Ali Ghanbari and Zhang Lingming, they introduced PraPR[95], which are the implementation of practical APR technology that runs on the JVM bytecode level. They believe PraPR can fix bugs for other JVM languages, such as Kotlin. Yazhini et al. implemented the MPPEngine[97] method in the Astor workspace by extending jGenProg. MPPEngine finding patches for six more bugs than jGenProg.

IV. EMPIRICAL FOUNDATIONS AND ARGUMENTATION

In ABF, there existed a series of fundamental problems explored, such as which bugs are more comfortable to be fixed, the accuracy, efficiency of the algorithm, and the practical application ability of the current algorithm. The foundation of empirical research is an essential step of ABF. Since different algorithms come from different researchers, it is still difficult to establish a standard evaluation criterion of algorithms in this field. Currently, the exploration of ABF mainly focuses on algorithm evaluation criteria, sources of candidate patch, and overfitting issues of test cases.

Monperrus [99] stated the content and algorithm evaluation criteria of the ABF research based on the test set. The core of their repair algorithm was to obtain high-quality patches under a given test set, whereas low-quality test sets were not the algorithm defects. Algorithm evaluation criteria should be algorithm comparison based on the same data set and the same defect type.

Martinez et al. [34] explored the existence of "redundancy assumption" in the form of empirical investigation. The so-called "redundancy assumption" referred to the fact that automatically generated patches must exist somewhere else in the program, and the repair algorithm was to reuse or combine code from other locations. The paper showed that the "redundancy assumption" did exist to some extent, but not completely. Also, Barr et al. [100] put forward the "plastic surgery hypothesis", like "redundancy assumption", that code in a patch could move from other parts of the program to the location of bug repair.

Qi et al. [101] analyzed the plausibility and correctness of patches by manually checking the repair results of the early algorithm on the dataset of 105 real C program bugs. The experimental results showed that due to the wrong experimental setting, 37 of the 55 repairable bugs reported in GenProg article [23] and 27 of the 54 in AE article [21] failed to pass all the test cases. Qi et al. also proposed GenProg-FL [102] and RSRepair [11]. GenProg-FL could accept the guidance of existing automatic bug localization methods. The authors believe that the lower NCP(Number of Candidate Patches) score meant the better repair effectiveness of a tool or method. RSRepair [11] replaced the genetic algorithm with a random search to verify the effectiveness of the genetic programming algorithm in the GenProg. Their experimental result proved that the genetic algorithm might not be the critical factor for generating patches. Smith et al. [103] studied the overfitting of the repair algorithm. They designed a controlled experiment by the introduction of bugs and patches by new developers. Their experiment analyzed the factors which influenced the effect of GenProg and RSRepair algorithm. Le et al. [104] revisited the overfitting problem and performed the first study on overfitting in semantic-based program repair by using IntroClass and Codeflaws benchmarks. Their paper showed that semantic-based APR techniques do indeed produce overfitting patches and substantiated that using multiple synthesis engines could mitigate overfitting in semantic-based ABF.

Furthermore, Yu et al. [105] analyzed the overfitting problems in ABF, identified two kinds of overfitting issues, and defined three kinds of overfitting patches. They also proposed UnsatGuided to alleviate the overfitting problems in

ABF techniques. It used additional automatically generated tests to strengthen the repair constraint. The empirical evaluation based on Nopol [47] and EvoSuite [106] showed that UnsatGuided was effective in alleviating overfitting issue of regression introduction for 16/19 bugs from Defects4J.

To verify each generated patch, the automatic program repair (APR) tool needs to run the same test suite repeatedly. When the number of patches is large, this process is expensive. Guo et al.[107] used Dynamic Software Update (DSU) technology to accelerate the automatic repair technology of the program, their results show that less than 1% patches cannot be dynamically updated using the builtin DSU ability of JVM, and DSU based validation leads to potentially harmful inconsistency in only 16 of 1,897,518 patches.

V. Challenge

A. Bug Fixing Time Is Very Different

Predicting bug fixing time can help developers better estimate software maintenance efforts and manage software systems. It is necessary to obtain an understanding of bug fixing time for improving the process of software maintenance. Zhang et al. [108] presented an approach based on Monte Carlo method [109] for estimating the total time required for ABF from historical data. They sampled 90% of the maintenance data as the training set and used Monte Carlo simulations to get the time of bug repair for the rest of 10% data. Weiss et al. [110] used the kNN-based model to search for similar earlier reports to predict the fixing time. In literature [5], a successful fixing needed 96 minutes and cost an average of $7.32 across eight open-source programs. In literature [36], they repaired a bug taking 3.8 minutes on average whereas method based on genetic programming repair took 6 minutes on average on the Coreutils programs along with real bugs. We can find that the time of fixing bugs is different.

B. Bug Fixing May Lead To New Bugs

Sometimes, the process of bug repair will often introduce new bugs. Gu et al. [111] found that bug fixing often failed to repair a bug or created new bugs and wrong fixing comprised about 9% of bugs by investigating bug databases of the Ant, AspectJ, and Rhino projects. Jin et al. [112] conducted a detailed characteristic study on incorrect bug fixing from large operating system code. They introduced incorrect bug repair as a new bug by investigating the mistake patterns and the likely human reasons during bug fixing. Nguyen et al. [113] discovered that 17% to 45% of total repairs were recurring bug repairs.

VI. Conclusion and Future Work

To better understand the bug fixing techniques, this paper systematically analyzes the research achievements made in the APR field in recent years. We also collect the commonly used defect benchmarks and all the available tools that are frequently used in the field of ABF. There are several breakthroughs in ABF techniques still to be done in the future. In future work, developers will design novel fitness functions for evolutionary computation and apply them to the area of fixing bugs in order to reduce the effort required to repair bugs automatically. They will search for more suitable constraint conditions of the programs and solve these constraint conditions in order to generate patches or variants for suggesting an optimal repair. Mixing different approaches are also an effective way to improve the quality of bug fixing.

ABF techniques will address the ambitious challenge of automatically fixing bugs in software. However, there are still significant research challenges in the automatic fixing of real bugs so that the ABF approach has not been successfully applied to the industry. In order to further improve the application value of this field, researchers should focus on the real-world application of software automatic repair in the practical industry.

Acknowledgment

This work was partially supported by the National Nature Science Foundation of China (No. 61602154, No. 61772173), the Fundamental Research Funds for the Provincial Universities of Henan University of Technology (No. 2016QNJH28), Science and Technology Program of Henan Province (No. 172102210216), and Key scientific research projects of Henan universities(No. 18B520013).

References

[1] W. E. Wong, X. Li, and P. A. Laplante, “Be more familiar with our enemies and pave the way forward: A review of the roles bugs played in software failures,” Journal of Systems and Software, vol. 133, pp. 68–94, Nov. 2017.

[2] C. Le Goues, S. Forrest, and W. Weimer, "Current Challenges in Automatic Software Repair," Software Quality Journal, vol. 21, no. 3, 2013, pp. 421-443.

[3] L. Gazzola, D. Micucci, and L. Mariani, "Automatic Software Repair: A Survey," IEEE Transactions on Software Engineering, 2017, pp. 34-67.

[4] M. Monperrus, "Automatic Software Repair: A Bibliography," Acm Computing Surveys, vol. 51, no. 1, 2017, pp. 1-24.

[5] C. L. Goues, T. V. Nguyen, S. Forrest, and W. Weimer, "GenProg: A Generic Method for Automatic Software Repair," IEEE Transactions on Software Engineering, vol. 38, 2012, pp. 54-72.

[6] F. Demarco, J. Xuan, D. L. Berre, and M. Monperrus, "Automatic Repair of Buggy If Conditions and Missing Preconditions with SMT," Proc. International Workshop on Constraints in Software Testing, Verification, and Analysis (CSTVA), 2014, pp. 30-39.

[7] A. Arcuri and X. Yao, "Coevolving Programs and Unit Tests from Their Specification," Proc. International Conference on Automated Software Engineering(ASE), 2007, pp. 397-400.

[8] S. Forrest, T. V. Nguyen, W. Weimer, and C. L. Goues, "A Genetic Programming Approach to Automated Software Repair," Proc. 11th Annual Conference on Genetic and Evolutionary Computation (GECCO), 2009, pp. 947-954.

[9] W. Weimer, S. Forrest, C. L. Goues, and T. V. Nguyen, "Automatic Program Repair with Evolutionary Computation," Communications of the ACM, vol. 53, no. 5, 2010, pp. 109-116.

[10] Y. Wei et al., "Automated Fixing of Programs with Contracts,"Proc. 19th International Symposium on Software Testing and Analysis(ISSTA), 2010.

[11] Y. Qi et al., "The Strength of Random Search on Automated Program Repair," Proc. International Conference on Software Engineering (ICSE), 2014, pp. 254-265.

[12] M. Wen et al., "Context-Aware Patch Generation for Better Automated Program Repair," Proc. 40th International Conference on Software Engineering (ICSE), 2018.

[13] X. B. D. Le, D. Lo, and C. L. Goues, "History Driven Program Repair," Proc. 23rd International Conference on Software Analysis, Evolution and Reengineering (SANER), 2016, pp. 213-224.

[14] A. Arcuri and X. Yao, "A Novel Co-Evolutionary Approach to Automatic Software Bug Fixing," Proc. IEEE World Congress on Computational Intelligence(IEEE WCCI), 2008, pp. 162-168.

[15] A. Arcuri, "On the Automation of Fixing Software Bugs," Proc. 30th International Conference on Software Engineering(ICSE) , 2008, pp. 1003-1006.

[16] W. Weimer, T. Nguyen, C. Le Goues, and S. Forrest, "Automatically Finding Patches Using Genetic Programming," Proc. 31st International Conference on Software Engineering(ICSE), 2009, pp. 364-374.

[17] W. E. Wong, V. Debroy, and B. Choi, "A Family of Code Coverage-Based Heuristics for Effective Fault Localization," Journal of Systems and Software, vol. 83, no. 2, 2010, pp. 188-208.

[18] V. Debroy and W. E. Wong, "Using Mutation to Automatically Suggest Fixes for Faulty Programs," Proc. 3th International Conference on Software Testing, Verification and Validation(ICST), 2010, pp. 65-74.

[19] V. Debroy and E. W. Wong, "Combining mutation and fault localization for automated program debugging," Journal of Systems and Software, vol. 90, pp. 45–60, Apr. 2014.

[20] W. E. Wong, V. Debroy, R. Gao, and Y. Li, "The DStar Method for Effective Software Fault Localization," IEEE Transactions on Reliability, vol. 63, no. 1, pp. 290–308, Nov. 2014.

[21] W. Weimer, Z. P. Fry, and S. Forrest, "Leveraging Program Equivalence for Adaptive Program Repair: Models and First Results," Proc. Automated Software Engineering(ASE), 2013.

[22] E. Fast, C. Le Goues, S. Forrest, and W. Weimer, "Designing Better Fitness Functions for Automated Program Repair," Proc. 12th Annual Conference on Genetic and Evolutionary Computation (GECCO), 2010, pp. 965-972.

[23] C. L. Goues, M. Dewey-Vogt, S. Forrest, and W. Weimer, "A Systematic Study of Automated Program Repair: Fixing 55 out of 105 Bugs for 8 Each," Proc. International Conference on Software Engineering(ICSE), 2012, pp. 3-13.

[24] D. Engler et al., "Bugs as Deviant Behavior: A General Approach to Inferring Errors in Systems Code," Proc. 18th ACM Symposium on Operating Systems Principles(SOSP), 2001, pp. 57-72.

[25] S. H. Tan, H. Yoshida, M. R. Prasad, and A. Roychoudhury, "Anti-Patterns in Search-Based Program Repair," Proc. 24th International Symposium on Foundations of Software Engineering(FSE), 2016, pp. 727-738.

[26] M. Martinez et al., "Automatic Repair of Real Bugs in Java: A Large-Scale Experiment on the Defects4j Dataset," Empirical Software Engineering, vol. 22, no. 4, 2017, pp. 1936-1964.

[27] Y. Xiong et al., "Precise Condition Synthesis for Program Repair," Proc. 39th International Conference on Software Engineering (ICSE), 2017, pp. 416-426.

[28] J. Jiang et al., "Shaping Program Repair Space with Existing Patches and Similar Code," Proc. International Symposium on Software Testing and Analysis (ISSTA), 2018, pp. 298-309.

[29] J. Jiang, Y. Xiong, and X. Xia, "A manual inspection of Defects4J bugs and its implications for automatic program repair," Science China Information Sciences, vol. 62, 2019, pp. 200102:1--200102:16.

[30] M. White et al., "Sorting and Transforming Program Repair Ingredients Via Deep Learning Code Similarities,"Proc. 26th International Conference on Software Analysis, Evolution and Reengineering (SANER), 2019, pp. 479-490.

[31] Z. Chen, S. Kommrusch, et al., "SEQUENCER: Sequence-to-Sequence Learning for End-to-End Program Repair," IEEE Transactions on Software Engineering, 2019.

[32] K. Liu et al., "Lsrepair: Live Search of Fix Ingredients for Automated Program Repair," Proc. 25th Asia-Pacific Software Engineering Conference (APSEC), 2018, pp. 658-662.

[33] A. Koyuncu, K. Liu, et al., "FixMiner: Mining relevant fix patterns for automated program repair," Empirical Software Engineering, vol. 25, 2020, pp. 1980-2024.

[34] M. Martinez, W. Weimer, and M. Monperrus, "Do the Fix Ingredients Already Exist? An Empirical Inquiry into the Redundancy Assumptions of Program Repair Approaches," Proc. 36th International Conference on Software Engineering(ICSE), 2014, pp. 492-495.

[35] S. Mechtaev et al., "Semantic Program Repair Using a Reference Implementation," Proc. 40th International Conference on Software Engineering (ICSE), 2018, pp. 129-139.

[36] H. D. T. Nguyen, D. Qi, A. Roychoudhury, and S. Chandra, "Semfix: Program Repair Via Semantic Analysis," Proc. International Conference on Software Engineering (ICSE), 2013, pp. 772-781.

[37] D. Gopinath, M. Z. Malik, and S. Khurshid, "Specification-Based Program Repair Using SAT," Proc. Tools and Algorithms for the Construction and Analysis of Systems International Conference (TACAS), 2011.

[38] H. Samirni et al., "Automated Repair of Html Generation Errors in Php Applications Using String Constraint Solving," Proc. International Conference on Software Engineering(ICSE), 2012, pp. 277-287.

[39] B. Demsky et al., "Inference and Enforcement of Data Structure Consistency Specifications," Proc. International Symposium on Software Testing and Analysis(ISSTA), 2006, pp. 233-244.

[40] M. D. Ernst, J. Cockrell, W. G. Griswold, and D. Notkin, "Dynamically Discovering Likely Program Invariants to Support Program Evolution," IEEE Transactions on Software Engineering, vol. 27, no. 2, 2001, pp. 99-123.

[41] P. Yu et al., "Code-Based Automated Program Fixing," Proc. 26th Automated Software Engineering(ASE), 2011, pp. 392-395.

[42] B. Daniel, T. Gvero, and D. Marinov, "On Test Repair Using Symbolic Execution," Proc. 19th International Symposium on Software Testing and Analysis(ISSTA), 2010, pp. 207-218.

[43] M. Z. Malik, K. Ghori, B. Elkarablieh, and S. Khurshid, "A Case for Automated Debugging Using Data Structure Repair," Proc. Automated Software Engineering(ASE), 2009, pp. 620-624.

[44] F. S. Ocariza, K. Pattabiraman, and A. Mesbah, "Vejovis: Suggesting Fixes for Javascript Faults," Proc. International Conference on Software Engineering(ICSE), 2014, pp. 837-847.

[45] S. Mechtaev, J. Yi, and A. Roychoudhury, "Directfix: Looking for Simple Program Repairs," Proc. 37th International Conference on Software Engineering(ICSE), 2015, pp. 448-458.

[46] S. Jha, S. Gulwani, S. A. Seshia, and A. Tiwari, "Oracle-Guided Component-Based Program Synthesis," Proc. International Conference on Software Engineering(ICSE), 2010, pp. 215-224.

[47] J. Xuan et al., "Nopol: Automatic Repair of Conditional Statement Bugs in Java Programs," IEEE Transactions on Software Engineering, vol. 43, no. 1, 2017, pp. 34-55.

[48] T. Durieux and M. Monperrus, "Dynamoth: Dynamic Code Synthesis for Automatic Program Repair," Proc. 11th International Workshop on Automation of Software Test(AST), 2016, pp. 85-91.

[49] Y. Ke, K. T. Stolee, C. Le Goues, and Y. Brun, "Repairing Programs with Semantic Code Search (T)," Proc. International Conference on Automated Software Engineering(ASE), 2015, pp. 295-306.

[50] K. T. Stolee, S. Elbaum, and D. Dobos, "Solving the Search for Source Code," ACM Transactions on Software Engineering and Methodology, vol. 23, no. 3, 2014, pp. 1-45.

[51] G. Le Claire, N. Holtschulte, E. K. Smith, et al., "The ManyBugs and IntroClass Benchmarks for Automated Repair of C Programs," IEEE Transactions on Software Engineering, vol. 41, 2015, pp. 1236-1256.

[52] J. Campos, A. Riboira, A. Perez, and R. Abreu, "Gzoltar: An Eclipse Plug-in for Testing and Debugging," Proc. Automated Software Engineering(ASE), 2012, pp. 378-381.

[53] S. Bhatia, P. Kohli, and R. Singh, "Neuro-Symbolic Program Corrector for Introductory Programming Assignments," Proc. 40th International Conference on Software Engineering(ICSE), 2018, pp. 60-70.

[54] X.-B. D. Le et al., "Jfix: Semantics-Based Repair of Java Programs Via Symbolic Pathfinder," Proc. 26th International Symposium on Software Testing and Analysis, 2017.

[55] C. S. Păsăreanu et al., "Symbolic Pathfinder: Integrating Symbolic Execution with Model Checking for Java Bytecode Analysis," Automated Software Engineering(ASE), vol. 20, no. 3, 2013, pp. 391-425.

[56] M. Soto and C. Le Goues, "Using a Probabilistic Model to Predict Bug Fixes," Proc. International Conference on Software Analysis, Evolution and Reengineering (SANER), 2018, pp. 221-231.

[57] Q. Gao et al., "Fixing Recurring Crash Bugs Via Analyzing Q&a Sites," Proc. International Conference on Automated Software Engineering (ASE), 2015, pp. 307-318.

[58] S. H. Tan and A. Roychoudhury, "Relifix: Automated Repair of Software Regressions," Proc. International Conference on Software Engineering (ICSE), 2015.

[59] M. Böhme and A. Roychoudhury, "Corebench: Studying Complexity of Regression Errors," Proc. International Symposium on Software Testing and Analysis (ISSTA), 2014, pp. 105-115.

[60] S. Sidiroglou-Douskos, E. Lahtinen, F. Long, and M. Rinard, "Automatic Error Elimination by Horizontal Code Transfer across Multiple Applications," ACM Sigplan Notices, vol. 50, no. 6, pp. 43-54, 2015.

[61] C. Liu, J. Yang, L. Tan, and M. Hafiz, "R2fix: Automatically Generating Bug Fixes from Bug Reports,"Proc. IEEE 6th International Conference on Software Testing(ICST), 2013, pp. 282-291.

[62] A. Koyuncu, K. Liu, et al., "iFixR: bug report driven program repair," Proc. 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE), 2019.

[63] D. Kim, J. Nam, J. Song, and S. Kim, "Automatic Patch Generation Learned from Human-Written Patches," Proc. International Conference on Software Testing(ICST), 2013, pp. 802-811.

[64] S. Kaleeswaran, V. Tulsian, A. Kanade, and A. Orso, "Minthint: Automated Synthesis of Repair Hints," Proc. International Conference on Software Engineering (ICSE), 2014, pp. 266-276.

[65] B. Cornu, T. Durieux, L. Seinturier, and M. Monperrus, "Npefix: Automatic Runtime Repair of Null Pointer Exceptions in Java," Computer ence, 2015.

[66] R. Gupta, S. Pal, A. Kanade, and S. Shevade, "DeepFix: Fixing Common C Language Errors by Deep Learning," Proc. Association for the Advancement of Artificial Intelligence(AAAI), 2017, pp. 1345-1351.

[67] X.-B. D. Le et al., "S3: Syntax-and Semantic-Guided Repair Synthesis Via Programming by Examples," Proc. 11th Joint Meeting on Foundations of Software Engineering(FSE), 2017, pp. 593-604.

[68] H. Do, S. Elbaum, and G. Rothermel, "Supporting Controlled Experimentation with Testing Techniques: An Infrastructure and Its Potential Impact," Empirical Software Engineering, vol. 10, 2005, pp. 405-435.

[69] R. Just, D. Jalali, and M. D. Ernst, "Defects4j: A Database of Existing Faults to Enable Controlled Testing Studies for Java Programs," Proc. International Symposium on Software Testing and Analysis (ISSTA) , 2014.

[70] S. H. Tan, J. Yi, S. Mechtaev, and A. Roychoudhury, "Codeflaws: A Programming Competition Benchmark for Evaluating Automated Program Repair Tools," Proc. 39th International Conference on Software Engineering Companion (ICSE-C), 2017, pp. 180-182.

[71] M. Böhme and A. Roychoudhury, "Corebench: Studying Complexity of Regression Errors," Proc. International Symposium on Software Testing and Analysis (ISSTA), 2014, pp. 105-115.

[72] M. Böhme et al., "Where Is the Bug and How Is It Fixed? An Experiment with Practitioners," Proc. 11th Joint Meeting on Foundations of Software Engineering(FSE), 2017, pp. 117-128.

[73] D. Lin, J. Koppel, A. Chen, and A. Solar-Lezama, "Quixbugs: A Multi-Lingual Program Repair Benchmark Set Based on the Quixey Challenge," Proc. ACM SIGPLAN International Conference on Systems, Programming, Languages, and Applications: Software for Humanity(SPLASH), 2017, pp. 55-56.

[74] H. Ye, M. Martinez, and M. Monperrus, "A Comprehensive Study of Automatic Program Repair on the QuixBugs Benchmark," CoRR, 2018.

[75] S. H. Tan, Z. Dong, X. Gao, and A. Roychoudhury, "Repairing Crashes in Android Apps," Proc. 40th International Conference on Software Engineering (ICSE), 2018.

[76] R. Saha et al., "Bugs.Jar: A Large-Scale, Diverse Dataset of Real-World Java Bugs," Proc. 15th International Conference on Mining Software Repositories(MSR), 2018, pp. 10-13.

[77] F. Madeiral, S. Urli, M. Maia, and M. Monperrus, "Bears: An Extensible Java Bug Benchmark for Automatic Program Repair Studies," Proc. 26th International Conference on Software Analysis, Evolution and Reengineering (SANER), 2019, pp. 468-478.

[78] D. A. Tomassi, N. Dmeiri, et al., "Bugswarm: Mining and continuously growing a dataset of reproducible failures and fixes,"Proc. 41th International Conference on Software Engineering(ICSE), 2019.

[79] T. Durieux and R. Abreu, "Critical Review of BugSwarm for Fault Localization and Program Repair," CoRR, 2019.

[80] D. A. Tomassi and C. Rubio-Gonz´alez, "A Note About Critical Review of BugSwarm for Fault Localization and Program Repair," CoRR, 2019.

[81] C. Cadar, D. Dunbar, and D. R. Engler, "KLEE: Unassisted and Automatic Generation of High-Coverage Tests for Complex Systems Programs," Proc. USENIX Symposium on Operating System Design and Implementation(OSDI), 2009, pp. 209-224.

[82] M. Martinez and M. Monperrus, "Astor: Exploring the Design Space of Generate-and-Validate Program Repair Beyond Genprog," Journal of Systems and Software, vol. 151, 2019, pp. 65-80.

[83] M. Martinez and M. Monperrus, "Astor: A Program Repair Library for Java," Proc. 25th International Symposium on Software Testing and Analysis(ISSTA), 2016, pp. 441-444.

[84] S. Urli, Z. Yu, L. Seinturier, and M. Monperrus, "How to Design a Program Repair Bot? Insights from the Repairnator Project," Proc. International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP), 2018, pp. 95-104.

[85] S. Mechtaev, J. Yi, and A. Roychoudhury, "Angelix: Scalable Multiline Program Patch Synthesis Via Symbolic Analysis," Proc. International Conference on Software Engineering (ICSE), 2016, pp. 691-701.

[86] L. Chen, Y. Pei, and C. A. Furia, "Contract-Based Program Repair without the Contracts," Proc. 32nd International Conference on Automated Software Engineering(ASE), 2017, pp. 637-647.

[87] T. Durieux, B. Cornu, L. Seinturier, and M. Monperrus, "Dynamic Patch Generation for Null Pointer Exceptions Using Metaprogramming," Proc. 24th International Conference on Software Analysis, Evolution and Reengineering (SANER), 2017, pp. 349-358.

[88] K. Liu, A. Koyuncu, D. Kim, and T. F. Bissyandé, "Avatar: Fixing Semantic Bugs with Fix Patterns of Static Analysis Violations,"Proc. 26th International Conference on Software Analysis, Evolution and Reengineering (SANER), 2019, pp. 1-12.

[89] L. Kui, K. Anil, K. Dongsun, and F. B. Tegawendé, "TBar: revisiting template-based automated program repair," Proc. 28th ACM SIGSOFT International Symposium on Software Testing and Analysis(ISSTA), 2019.

[90] T.-T. Nguyen, Q.-T. Ta, and W.-N. Chin, "Automatic Program Repair Using Formal Verification and Expression Templates," Proc. International Conference on Verification, Model Checking, and Abstract Interpretation(VMCAI), 2019.

[91] Q. Xin and S. P. Reiss, "Leveraging syntax-related code for automated program repair," Proc. 32nd International Conference on Automated Software Engineering (ASE), 2017.

[92] Q. Xin and S. P. Reiss, "Revisiting ssFix for Better Program Repair," CoRR, 2019.

[93] Q. Xin and S. P. Reiss, "Better Code Search and Reuse for Better Program Repair," Proc. 6th International Workshop on Genetic Improvement(GI), 2019.

[94] A. Afzal, M. Motwani, K. Stolee, Y. Brun, and C. Le Goues, "SOSRepair: Expressive Semantic Search for Real-World Program Repair," IEEE Transactions on Software Engineering, 2019, pp. 1-22.

[95] A. Ghanbari and L. Zhang, "PraPR: Practical Program Repair via Bytecode Mutation," Proc. 34th International Conference on Automated Software Engineering (ASE), 2019.

[96] Z. Chen, S. Kommrusch, et al., "SEQUENCER: Sequence-to-Sequence Learning for End-to-End Program Repair," IEEE Transactions on Software Engineering, 2019.

[97] Y. Venugopal, P. Quang-Ngoc, and L. Eunseok, "Modification Point Aware Test Prioritization and Sampling to Improve Patch Validation in Automatic Program Repair," Applied Sciences, vol. 10, no. 5, 2020.

[98] A. Aleti and M. Martinez, "E-APR: Mapping the Effectiveness of Automated Program Repair," CoRR, 2020.

[99] M. Monperrus, "A Critical Review of "Automatic Patch Generation Learned from Human-Written Patches": Essay on the Problem Statement and the Evaluation of Automatic Software Repair," Proc. 36th International Conference on Software Engineering(ICSE), 2014.

[100] E. T. Barr et al., "The Plastic Surgery Hypothesis," Proc. 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering(FSE), 2014, pp. 306-317.

[101] Z. Qi, F. Long, S. Achour, and M. Rinard, "An Analysis of Patch Plausibility and Correctness for Generate-and-Validate Patch Generation Systems," Proc. International Symposium on Software Testing and Analysis(ISSTA), 2015, pp. 24-36.

[102] Y. Qi, X. Mao, Y. Lei, and C. Wang, "Using Automated Program Repair for Evaluating the Effectiveness of Fault Localization Techniques," Proc. International Symposium on Software Testing and Analysis(ISSTA), 2013, pp. 191-201.

[103] E. K. Smith, E. T. Barr, C. Le Goues, and Y. Brun, "Is the Cure Worse Than the Disease? Overfitting in Automated Program Repair," Proc. Joint Meeting on Foundations of Software Engineering (ESEC/FSE), 2015, pp. 532-543.

[104] X. B. D. Le, F. Thung, D. Lo, and C. Le Goues, "Overfitting in Semantics-Based Automated Program Repair," Empirical Software Engineering, vol. 23, no. 5, 2018, pp. 3007-3033.

[105] Z. Yu et al., "Alleviating Patch Overfitting with Automatic Test Generation: A Study of Feasibility and Effectiveness for the Nopol Repair System," Empirical Software Engineering, vol. 24, no. 1, 2019, pp. 33-67.

[106] G. Fraser and A. Arcuri, "Evosuite: Automatic Test Suite Generation for Object-Oriented Software,"Proc. 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering, 2011.

[107] R. Guo, T. Gu, Y. Yao, F. Xu, and X. Ma, "Speedup Automatic Program Repair Using Dynamic Software Updating," Proc. 11th Asia-Pacific Symposium on Internetware, 2019.

[108] H. Zhang, L. Gong, and S. Versteeg, "Predicting Bug-Fixing Time: An Empirical Study of Commercial Software Projects," Proc. International Conference on Software Engineering(ICSE), 2013, pp. 1042-1051.

[109] L. Devroye, Non-Uniform Random Variate Generation. Springer-Verlag, New York, 1990.

[110] C. Weiss, R. Premraj, T. Zimmermann, and A. Zeller, "How Long Will It Take to Fix This Bug?, " Proc. 4th International Workshop on Mining Software Repositories(MSR), 2007, pp. 1-8.

[111] Z. Gu, E. T. Barr, D. J. Hamilton, and Z. Su, "Has the Bug Really Been Fixed?," Proc. International Conference on Software Engineering(ICSE), 2010, pp. 55-64.

[112] G. Jin et al., "Automated Atomicity-Violation Fixing," Proc. Programming Language Design and Implementation(PLDI), 2012, pp. 389-400.

[113] T. T. Nguyen et al., "Recurring Bug Fixes in Object-Oriented Programs," Proc. International Conference on Software Engineering(ICSE), 2010, pp. 315-324.

Security of Edge Computing Based on Trusted Computing

Bin Ma
School of Information Engineering
North China University of Water Resources and Electric Power
Zhengzhou,China
mabin@ncwu.edu.cn

Ziying Ye
School of Information Engineering
North China University of Water Resources and Electric Power
Zhengzhou,China
sweet9394@126.com

Xufang Zhang
School of Information Engineering
North China University of Water Resources and Electric Power
Zhengzhou,China
zhangxufang@ncwu.edu.cn

Jiajing Chen
School of Information Engineering
North China University of Water Resources and Electric Power
Zhengzhou,China
1415931869@qq.com

Yang Zhou
School of Information Engineering
North China University of Water Resources and Electric Power
Zhengzhou,China
335121690@qq.com

Qing Xia
School of Management
Guangdong University of Education
Guangzhou,China
xiaqing@gdei.edu.cn

***Abstract*—With the development of information communication technology and Internet of Things technology, the number of devices connected to the network and the amount of data generated are exponentially increasing, thus resulting in a series of new application scenarios. The traditional centralized cloud computing big data processing model can no longer content the current data growth needs. Thus, born edge computing, a new computing model that migrates some or all of the computing tasks of the original cloud computing center to near the data source, gradually received extensive attention from all walks of life. According to the current cloud computing and edge computing application scenarios, analyze some security threats faced by edge computing in applications. Analyze and organize the security part of the existing edge computing reference architecture, and refine the edge computing security protection architecture and principles. By introducing trusted computing and blockchain technology, the credibility and adaptability of the edge computing security protection system is increased, making it more adaptable to the current practical application scenarios.**

Keywords—Edge computing, safety protection, Trusted Computing, Blockchain

I. INTRODUCTION

Nowadays, with the development of information communication technology and Internet of Things technology, the number of devices connected to the network and the amount of data generated are exponentially increasing, thus resulting in a series of new application scenarios. Thus born edge computing, a new computing model that migrates some or all of the computing tasks of the original cloud computing center to near the data source, gradually received extensive attention from all walks of life, related companies and related open source platforms have also gradually developed. A series of edge computing reference architectures have been introduced, and they all attach great importance to data security. However, there is no independent security framework to systematically discuss the security of edge computing, which is not conducive to its popularize and application. Trusted computing as a new security method, combining trusted computing with edge computing, used to solve security problems in edge computing. It may enable edge computing to break through security bottlenecks, thereby better development.

II. EDGE COMPUTING SECURITY STATUS

Since the advent of cloud computing in 2005, great changes have taken place in software and applications. The cloud computing data center is the computing core of the network. All computing power, data storage, network bandwidth, power distribution, etc. in the entire computing network are deployed around the data center. Although it is a very effective way to put all the computing tasks in the data center for calculation, with the explosive growth of device data, the data transmission speed of the cloud computing framework has become a bottleneck to improve computing power. With the development of the Internet of Things, some IoT devices require very short response times, some may involve private data, and some may generate a large amount of duplicate data and increase the network load. Cloud computing alone cannot meet these needs. We need to process data directly on network edge devices.

Edge computing, after distributed computing, grid computing, and cloud computing, is considered a new type of computing paradigm that performs computing at the edge of the network. Its core idea is that computing is closer to the source of data. According to the definition of Edge Computing Consortium (ECC), edge computing refers to an open platform that is close to the edge of the network or the source of the data and integrates core capabilities such as network, computing, storage, and applications. Meet the key needs of industry digital in agile connections, real-time business, data optimization, application intelligence, security and privacy. Generally speaking, it is to sink the cloud computing storage capacity to the edge nodes of the network, use distributed computing and storage to directly process locally or solve

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00030

specific business needs, so as to meet the hard requirements of the new business format for high bandwidth and low latency.

The development of edge computing technology and industry applications is in its infancy, and it is currently being piloted in commercial and civil fields. With the establishment of ECC, edge computing architectures 1.0 to 3.0 were successively released [1-3], involving various functional domains, it has covered all aspects of edge computing such as hierarchical division and functions, technology implementation and interaction, application architecture mode and deployment. However, the current main security protection capabilities of edge computing are based on the stacking of traditional security protection equipment and software. The main core protection concept is still to perform superimposed or serial protection around the network boundary. The Open Fog Consortium (OFC) is an industrial organization established at the end of 2015, which proposed the concept of fog computing. At the beginning of the establishment of OFC, the "*Open Fog Alliance Fog Computing Reference Architecture*" (OpenFog RA) [4] In this architecture, the concept, framework and application cases of fog computing were systematically introduced in detail. Because OpenFog RA is relatively long, and security-related content is scattered in different chapters, the systematic and organized structure of content is slightly insufficient. ETSI released the "*Introducing Technical White Paper on Mobile Edge Computing*" in 2014, which introduced the concept of mobile edge computing MEC and proposed a system abstract architecture diagram [5]. However, the MEC white paper mentions relatively few security parts, briefly introduces security challenges and ways to deal with them, and lacks a systematic discussion. The American Industrial Internet Consortium (IIC) released the white paper "*Introduction to Edge Computing in IIOT*" in 2018 [6]， During the construction of IIOT, it mainly refers to the end-to-end security framework of IISF, but when adopting the edge computing architecture, the endogenous security of the device and the network should also be considered, the security of the computing and network nodes should be monitored, version updates, attack isolation and post-attack recoverability should be considered.

III. Status of trusted computing applications

The concept of trusted computing was proposed by TCPA, but there is no clear definition, and the understanding of "trusted computing" is also different among its members. The definition of "trusted" by a trusted computing organization is that when an entity achieves a given goal, if its behavior is always as expected, then the entity is trusted. The definition of "credible" focuses on the predictability of its results, and requires meeting the requirements of availability and high reliability.

Compared with the traditional and traditional terminal security solutions based on virus protection, firewalls, and intrusion detection, the core idea of trusted computing is that an embedded microcomputer system such as TPM is embedded in the computer platform. It can solve many problems that could not be solved before. TPM actually adds a trusted third party to the computer system, and the system can be trusted by the trusted third party by measuring and constraining the system.

In his paper [7], Du Song mentioned the application of trusted computing technology in cloud computing security, and described the solutions of trusted computing in cloud computing from various aspects of cloud computing security. Liu Youqi mentioned in his paper [8] the big data encryption technology based on trusted computing, and carried out research on big data trusted encryption from many levels. Shen Hao mentioned in his paper [9] the research on the trusted computing system of the domestic basic hardware and software platform, and developed the trusted application system from the hardware and software platform.

IV. Application of Trusted Computing in Edge Computing

A. *Trusted computing technology can be used to control and track users who access data*

Users put their complete identification information in the edge computing system in order to use certain mechanisms to track and limit their behavior. Users use their user names and passwords on a trusted computing platform to complete their identity information when the user enters the system It is difficult to hide the identification information, so the system will automatically check and record the identification information, mainly some information of the visitors who enter the system. The trusted computing platform can enable the edge computing system to control the visitor's behavior through the tracking mechanism.

B. *Ensure the safety and stability of network data*

Research on network data security and data stability has always been a very important research direction in the process of studying cloud computing, and powerful trusted computing technology has completely changed cloud computing in many ways. Trusted computing technology uses TPM technology to protect the relevant information it transmits. This increases the complexity of attacks on transmitted data and prevents user data from being altered or violated. In addition, the user or network application must complete the TPM authentication process before accessing the data. In this way, the probability of a vulnerability during data access is greatly reduced.

C. *Use trusted computing TPM to encrypt and authenticate the server and client*

When managing cloud platforms, user access and authentication are the tasks that must be done before accessing data. When users access data, they first need TPM authentication. On the one hand, there is a huge amount of user information in the process of authenticating cloud platform data. When a user browses through relevant webpages, the cloud platform needs to control the content displayed by the user's browser. The specific reason for the control is to ensure the security of the information and to ensure that its use is within the normal range. The secure cloud platform uses TPM's trusted technology authentication to provide users with a master key. The master key implements its own system. On the other hand, when adding a password to the relevant operation data on the cloud platform, TPM trusted technology provides a key for encrypting user data information and stores the key itself. The security of the password has been strengthened and improved, and the security of data access has also been guaranteed [10].

D. *Trusted computing support for compliance*

Due to the opacity of the cloud computing environment, users are very concerned about the security of cloud storage data. Generally, a reliable cloud server is installed to implement access control policies and perform monitoring or

auditing, and provide data owners with evidence of compliance.

V. THE THREAT TO EDGE COMPUTING

According to the edge computing reference architecture model jointly released by ECC and AII, the edge computing architecture is vertically divided into application domain, data domain, network domain, and device domain. The security threat to the entire edge computing system mainly comes from these domains and superposition factors. Each region's threats and risks have their own characteristics.

1) There are many edge computing terminals and the structure is complex: Due to the characteristics of the edge computing model, there are many access endpoints in the edge network, and they are distributed in different geographical locations. The application characteristics and security protection systems of different node terminals are different, resulting in the failure of edge terminal protection measures.

2) Network instability: Due to the characteristics of edge computing, the situation of edge nodes is different, the resources of edge nodes are limited, and the security protection measures of some nodes cannot meet the requirements, making it difficult for some nodes to defend against external network attacks.

3) Data and storage: According to the characteristics of the edge side access, the amount of data input from the edge side is large and the format is complex. There is a risk of data leakage in various processes such as data generation, transmission, storage, analysis, calculation, and sharing of the entire system. In addition, there are storage media on the edge side and the center side. The storage media and storage technology are different, and the corresponding security difference is large.

4) Management and access: The edge computing system is large in scale, and there are many edge network access and interactive equipment systems. It is necessary to build a trusted edge network through identity authentication. At the same time, the security management system and personnel operation management need to be related to management and control, using trusted technology to track and management.

VI. EDGE COMPUTING SECURITY PROTECTION SYSTEM

The system design of the edge computing security architecture needs to be divided around the technology of the edge computing application architecture. Edge computing presents a human-like working mechanism in actual application scenarios. The cloud computing center is responsible for global resource management, task coordination, and security overall control. Fog computing serves as the central nerve and serves as a regional computing, storage, and job coordination center, responsible for local task coordination, resource allocation and security control. The edge terminal interacts with actual users and other network elements and is responsible for task execution and information collection and reporting. Security threats run through the entire process node and network. According to the characteristics of the domain, network space and physical space where they are located, design and deploy security equipment and configure personalized security rules. Design and implement a deep-customized security plan for the entire defense field.

The design and implementation of the security defense system of the edge computing system, covering all levels of the edge computing application system. There are different attributes and safety requirements between different levels, which is an important reference and basis for the differentiated design of safety system schemes. In order to achieve the standardization and unification of the edge computing security protection system architecture, in general, unified situational awareness, unified process orchestration, unified identity authentication and authority management, unified security operation and maintenance and emergency response are required. Maximize the security and reliability of the entire edge computing network system [11]. The entire safety protection system is shown in Figure 1.

- Application layer security mainly includes access to whitelists, malicious attack prevention, security detection and response, application security audits, software hardening and patching, and access behavior supervision. Due to the massive heterogeneous access of terminals and the variety of services, the traditional IT security authorization model is no longer applicable, and it is often necessary to use a minimum authorization security model to manage applications and access permissions.
- Data layer security mainly includes data tamper resistance, data access control, data privacy protection, data encryption, data leakage prevention, data isolation and destruction, etc. Data encryption is the encryption of the entire data process, including storage, transmission, sharing, etc. Among them, data leakage prevention needs to consider that even if the hardware device is stolen, it will not disclose any information.
- Network layer security includes network security isolation, IPSee, firewall, intrusion detection and protection, DDoS protection, encrypted communication, etc. Among them, DDoS protection is crucial in the Internet of Things and edge computing.
- Node layer security mainly includes remote node upgrade, hardware security hardening, software security hardening, antivirus, vulnerability scanning, and lightweight trusted computing. The safe and reliable remote upgrade can complete the repair of vulnerabilities and patches in time to avoid system failure after the upgrade. Lightweight trusted computing is used to calculate simple IoT devices with limited CPU and storage resources to solve the most basic trusted problems.
- Resource layer security mainly includes resource access control, database security protection, cloud host cloud terminal security, and virtual machine resource security.
- Physical layer security mainly includes physical access control, anti-theft, waterproof, fire and lightning protection in the computer room, and temperature and humidity control. The defense at the physical level needs to be determined in conjunction with the specific realities in specific areas.

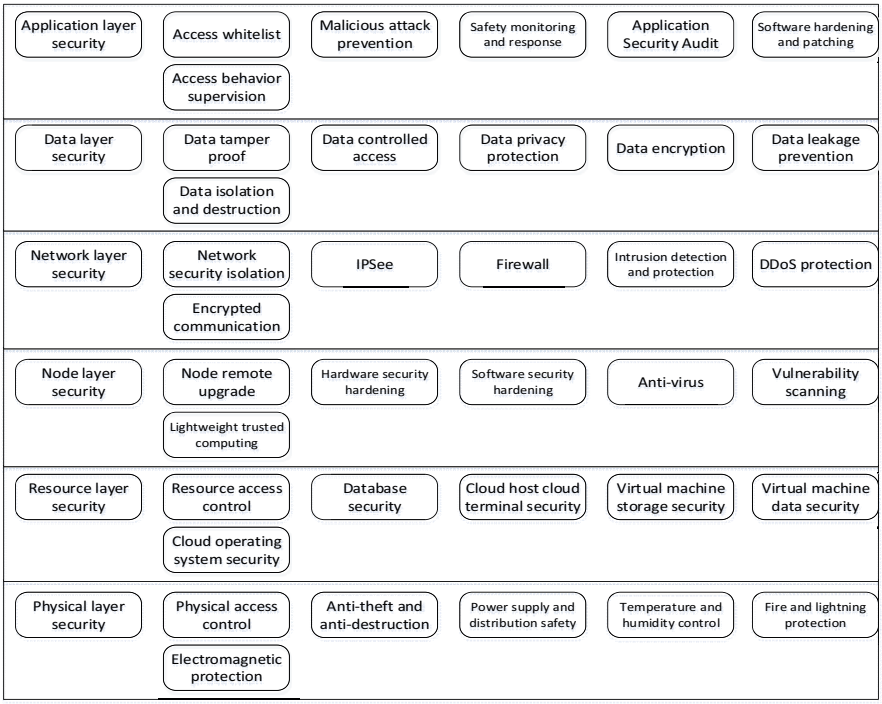


Fig. 1. Edge computing security protection diagram.

- As the security infrastructure of the entire edge computing system, it is necessary to strengthen the security protection of the relevant parts of the password and the management system. Through quantum cryptography, secure multi-party computing, and zero-knowledge proofs, we respond to new technology changes and the security risks introduced by quantum technology. The PKI public key infrastructure of the edge computing system adopts blockchain technology to realize the management of certificates, keys and the load balance of the cryptographic machine. With the help of the distributed consensus accounting technology of the blockchain, hidden communication and auditing, the security of the trusted edge network is guaranteed, and the security risk of single point of failure is reduced.

VII. EDGE COMPUTING TRUSTED PROTECTION MECHANISM

On the basis of the global collaborative security protection of the edge computing, the trusted security technology and the characteristics of the blockchain are used to build an edge security trusted protection network. These include device trust, regional trust, global trust, and shared trust. Use the blockchain's consensus accounting technology to achieve the robust characteristics of system security protection and prevent the destruction of security protection capabilities caused by single-point failures or DDoS attacks of traditional security protection technologies.

A. Device Trust

Device trust is to realize the encryption of a single device by installing a TPM chip on the edge device. It can control single device login, application. And implement functions such as secure I/O and remote certification, so that a single device can be trusted. Encrypt authorized access to control the behavior of edge device managers. However, virtualization is currently the key technology in the cloud model [12-13]. Virtualization technology can greatly improve the utilization rate of server hardware, and provide a highly scalable platform for applications through virtualization. Virtual machine technology is an important field of application of virtualization technology. It can significantly improve the efficiency of server usage and the security of user virtual machines in cloud computing mode, which is the premise and foundation of user security. Since multiple users are in a shared resource mode among cloud servers, it is possible to bind many different virtual resources to the same physical resources, and if there are some security vulnerabilities in the virtualization software in the cloud platform, then The user's data may be accessed by others, which cannot guarantee the security of the user's virtual machine. For the above problems, you can use the open source virtual machine monitor system Xen platform. On the cloud server, a physical TPM is mapped into multiple virtual TPMs. The virtual TPM can provide a dedicated root of trust for each virtual machine. Use Xen platform to implement virtual TPM, users can build a trusted virtual environment based on virtual TPM. There is a one-to-one correspondence between virtual TPM instances and user virtual machines, which can provide users with functions consistent with physical TPM such as binding and key storage.

B. Regional credibility

Using the private chain characteristics of the blockchain can create a secure and trusted network for the edge computing area. Private chain can easily modify the rules to facilitate the access of equipment. The nodes in the private chain can be better connected. Faults in the chain can be quickly repaired by manual intervention, with the help of multiple identity authentication or zero-knowledge proof technology to ensure that heterogeneous edge terminals (mobile phones, cars, sensors, etc.) accessing the edge

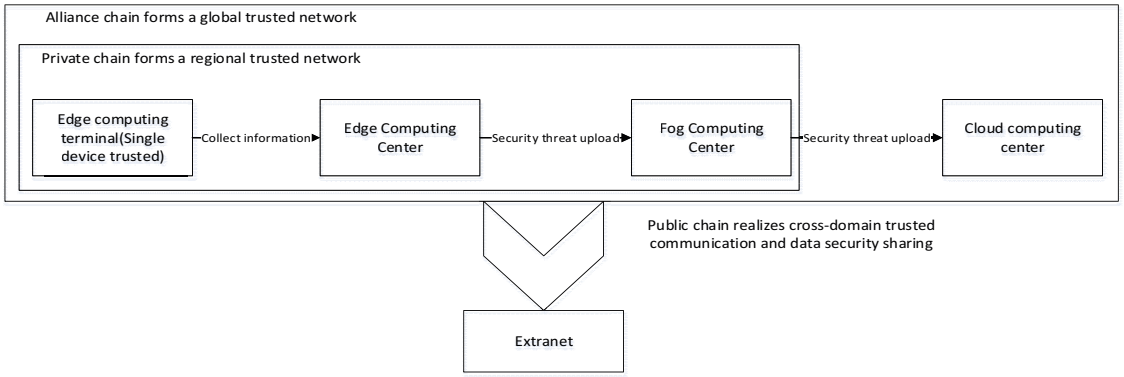


Fig. 2. Trust mechanism for edge computing network space.

network are credible, enabling edge terminals to access the network Safety control in advance.

C. Global trust

Using the alliance chain and edge gateway of the blockchain, a global trusted network for edge computing can be realized, and the trusted resources of the edge network can be freely combined and arranged in the entire edge computing network to cope with different situations. Using the non-tamperable characteristics of the blockchain to achieve ongoing supervision and retrospective, so as to ensure the authenticity and integrity of the data. Use the P2P mechanism and encrypted communication mechanism of the blockchain to achieve end-to-end covert communication and secure sharing of important data in the edge computing network.

D. Shared trust

Use the public chain of the blockchain to achieve cross-domain trusted communication and secure data sharing between the edge computing network and the external Internet. In the public chain, program developers have no right to interfere with users, and the blockchain can protect users who use the program from attack. Through the integration of smart contracts and smart algorithms of the blockchain, it is possible to carry out independent intelligent security protection and risk response according to security threat scenarios.

VIII. CONCLUSION

Edge computing is currently in the exploration period of technology application and industrial development. Various international and domestic edge computing organizations and institutions have also launched white papers on edge computing architecture. According to the development of authoritative institutions and industries, it is predicted that cloud computing will develop in the direction of cloud computing centers, fog computing, edge computing, and intelligent computing. However, the lack of security protection architecture and system for edge computing may lead to greater risks in the development of the edge computing industry. This paper mainly analyzes the threats faced by edge computing and the current security architecture, and puts forward a protection mechanism based on trust and cooperates with blockchain technology to improve the security and reliability of edge computing. If a separate security framework for edge computing can be proposed according to this article, it will play a role in the healthy development of edge computing. Edge computing technology is not yet mature. Some new technologies for edge computing, such as edge caching [14] and transfer learning [15], have also proposed new solutions to the security problems of edge computing.

ACKNOWLEDGMENT

First of all, I want to thank my teacher, Pro. Ma, who helped me complete this paper and helped me when I was the most difficult. Secondly, I would like to thank my classmate Ms. C for helping me improve my English and help me point out several errors in the paper. Special thanks to the friends who supported me in your work, you gave me a lot of encouragement. Finally, I am indebted to my parents for their continuous support and encouragement.

This study was financially supported by the National Natural Science Foundation of China (No. 51304078), Henan Provincial Key Research Projects for Colleges and Universities (No. 18A520035).

REFERENCES

[1] Edge Computing Consortium,"Edge Computing Consortium White Paper," Edge Computing Consortium, Beijing, 2016.

[2] Edge Computing Consortium and Alliance of Industrial Internet," Edge Computing Reference Architecture 2.0," Edge Computing Consortium and Alliance of Industrial Internet, Beijing, 2017.

[3] Edge Computing Consortium and Alliance of Industrial Internet," Edge Computing Reference Architecture 3.0," Edge Computing Consortium and Alliance of Industrial Internet, Beijing, 2018

[4] OpenFog,"Reference architecture for fog computing," OpenFog Consortium,Princeton, N.J., USA, 2017.

[5] M. Patel, B. Naughtom, C. Chan, et al.,"Mobile-edge computing introductory technical white paper," European telecommunications Standards Institute(ET-SI),London, UK,2014.

[6] American Industrial Internet Consortium,"Introduction to edge computing in IIoT," Industrial Internet Consortium(IIC), Needham, mass,USA,2018.

[7] D. Song," Application of Trusted Computing Technology in Cloud Computing Security," Communication World,vol. 4,2020,pp.90-91.

[8] L. Youqi," Application Research of Big Data Encryption Technology Based on Trusted Computing,"Information Technology and Informatization,vol. 6,2019,pp.170-172.

[9] S. Hao," Application Research on Trusted Computing System of Domestic Basic Software and Hardware Platform,"Electronic Technology,vol. 5,2018,pp.19-21.

[10] Y.Shaoyu," Research on Key Technologies of Cloud Service Resource Security Protection Mechanism," Zhengzhou University,Zhengzhou,2013.

[11] D. Xingyu, L. Fei, C. Jie," Research on Security Protection System of Edge Computing," Communication Technology,vol. 1,2020,pp.201-209.

[12] M. Raykova,B. Vo,S. M. Bellovin, et al.,"Secure Anonymous Database Search,"Proceedings of the 2009 ACM Workshop on Cloud Computing Security,2009.

[13] J. Crampton,K. Martin,P. Wild,"On Key Assignment for Hierarchical Access Control,"Proc. of the 19th IEEE Computer Security Foundations Workshop-CSFW,2006.

[14] U. Drolia,K. Guo,J. Tan,et al.,"Cachier: Edge-caching for recognition applications,"Proceeding of the 37th IEEE International Conference on Distributed Computing Systems (ICDCS),Wahingtong,D.C.,USA,2017,pp.276-286.

[15] R. Sharma,Biookaghazadehs,B. Li,et al.,"Are existing knowledge transfer techniques effective for deep learning with edge devices? ,"Proceedings of 2018IEEE International Conference on Edge Computing (EDGE).,Washington, D. C., USA, 2018,pp. 42-49.

Image Quality Measurement by Probabilistic Principal Component Analysis

Hua-Wen Chang
School of Computer and Communication Engineering
Zhengzhou University of Light Industry
Zhengzhou,China
e-mail:changhuawen@126.com

Kai Chen
School of Computer and Communication Engineering
Zhengzhou University of Light Industry
Zhengzhou,China
e-mail:1023626617@qq.com

Xiao-Dong Bi
School of Computer and Communication Engineering
Zhengzhou University of Light Industry
Zhengzhou,China
e-mail: 2547670355@qq.com

Ming-Hui Wang
College of Computer Science
Sichuan University
Chengdu,China
e-mail:wangminghui@scu.edu.cn

***Abstract*—In order to evaluate the perceptual quality of images, a full-reference quality index, which is called principal component deviation (PCD), is presented. This research is motivated by the discovery that the filters learned by the probabilistic principal component analysis are closely resemble the neurons of the human visual system. These filters are used as a model of the visual system, which are trained on 100,000 color image patches of size 4×4 by probabilistic principal component analysis. The PCD relates the image quality with the deviation between two sets of features that extracted by the learned filters. Experimental results show that PCD has relatively low computational complexity and high correlation with subjective quality evaluations.**

Keywords- Image quality index, Full-reference, Probabilistic principal component analysis;

I. INTRODUCTION

Image quality measurement (IQM) is essential to many related researches and applications, including image acquisition, compression and transmission. Thus, IQM is a fundamental issue in the area of image processing. The peak signal-to-noise ratio (PSNR) is a traditional image quality index (IQI), which directly measures the signal errors. Unfortunately, this popular IQI is not consistent with the human visual perception. Since images are ultimately to be viewed by human beings, it is desirable to have image quality indexes (IQIs) that can predict the perceived visual quality as measured with human subjects. To this end, many researchers have developed sophisticated IQM models to achieve perceptual consistency in quality prediction by modelling physiological response properties of the human visual system (HVS). Among all these methods, structural similarity (SSIM) index[10,13] is quite attractive owing to its simplicity and excellent performance relative to old methods such as the PSNR and mean squared error (MSE). It is based on the hypothesis that the HVS is highly adapted for extracting structural information in images. Some recent researches[2,3] relate the sparse coding to IQM, and present methods by modelling the neurons in the visual cortex. Another recent IQI called visual saliency-induced index (VSI)[12] is based on the fact that some areas of an image can attract the most attention of the HVS.

However, most research only focus on the features of grayscale images. Color images are always transformed into grayscale images by means of color space transformations. Chromatic features that closely in correlation with visual perception are rarely involved in the research of IQM. Aiming at this problem, our research tries to design a method that can extract all the color information as well as the other features that are related to IQM. An ideal visual model for perceptual quality measurement should be closely related to the neural response properties of the HVS[1]. Under this premise, we attempt to find an appropriate visual model that can make use of all the color information to predict the image quality precisely. Our research found that a group of visual filters can be obtained by applying probabilistic principal component analysis (PPCA)[8,9] on color image samples. Moreover, the frequency and the orientation representations of these filters are similar to those of the neurons of the HVS[5]. Inspired by this discovery, we propose to make use of such a group of filters to simulate the visual processing in the designing of IQM algorithms.

In this paper, a new IQI is proposed to deal with IQM problem through simulating the information processing of the HVS by PPCA. The proposed index is names as principal component deviation (PCD). Firstly, a group of visual filters are learned from samples of natural images by an expectation maximization algorithm for PPCA. Then the image quality score is considered as the deviation between two sets of visual features that are acquired by the learned visual filters. The computation of the quality score for a given image consists of two stages: filter creation and deviation computation. In the first stage, a set of filters is obtained through PPCA, these filters are considered as a model of the HVS. In the second stage, the filtering results are used to compute the quality score.

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00031

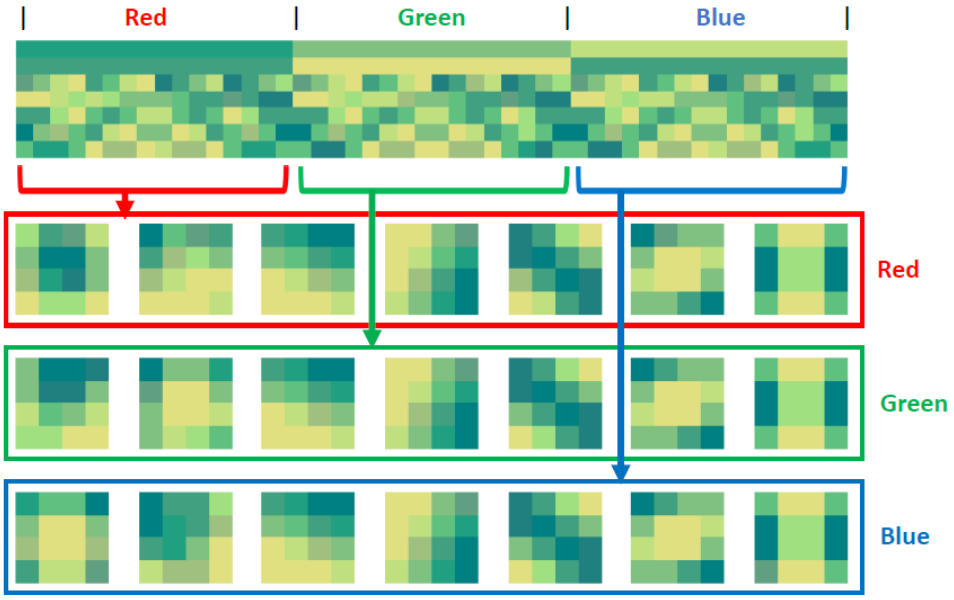


Figure 1. The matrix of principal component coefficients and corresponding filter banks.

The rest of this paper is organized as follows. In Section 2, the training steps of the filters are introduced. Section 3 talks about the computation details of the PCD. The experimental results and discussions are provided in Section 4 followed by the conclusions in Section 5.

II. Training of The Filters

The quality score of PCD can be obtained through two stages: training of the filters and deviation computation.

In the first stage, 100,000 color image patches of size 4×4 are randomly sampled from 100 natural images that come from the General-100 image dataset [6]. For these image samples, the mean value of each patch should be subtracted before the training process. To perform the PPCA on the samples, each image patch should be changed into a vector. Then, all the image patches can form a matrix of size 100,000×48. After that, we can perform PPCA using the expectation maximization algorithm and calculate the component coefficients. The resulting component coefficients are in the form of a matrix, which can project the image data from the high-dimensional coordinates to the low-dimensional principal component coordinates.

The number of the principal components (the dimensionality of the resulting coefficient vectors) is critical to the performance of the PCD. Through our experiments, we found that the proper number of principal components is 7. Thus, this training can result in a 7×48 matrix, as shown in the top of Fig. 1. The matrix of principal component coefficients can be changed into three groups of filters (blocks). Each row of the matrix has three sections (i.e., red, green and blue) corresponding to the three color channels of RGB images. Each section of one row can form a 4×4 block which is used as an image filter. Since the matrix has 7 row vectors, each group has 7 filters aiming to extract different features. Moreover, this matrix can also be considered as a visual model, each row of this matrix is one principal component, which can form three blocks of size 4×4. These blocks can be used as visual filters, as can be seen in Fig. 1, the spatial structure of these Gabor-like filters is similar to the receptive field structure of visual neurons. All the filters can be divided into three groups that can deal with the information of the RGB channels, for example, the filters in the red rectangle are especially used to extract features from the red channel of a given image. Each group contains 7 filters aiming to extract the features in different sub-bands.

In order to compute the quality score for a given image, these filters will be applied to the reference and test images in the second stage.

III. Deviation Computation

Firstly, every reference and test image should be subtracted the mean of each color channel by.

$$e\left(I_c\right) = I_c - \mu\left(I_c\right) \tag{1}$$

where c∈{red,green,blue} denotes the color channel of a given image I, the function μ(.) calculates the mean pixel value of each color channel.

In the first stage, a matrix is trained from image samples, moreover, this matrix can be divided into three groups of filters corresponding to three color channels of a RGB image. Then, we will make use of these filters for feature extraction.

Let X_c and Y_c denote the reference and test image, then we use the three groups of filters to extract features from the color channels by

$$\begin{aligned} FX_{(c,j)} &= e\left(X_c\right) \otimes f_{(c,j)} \\ FY_{(c,j)} &= e\left(Y_c\right) \otimes f_{(c,j)} \end{aligned} \tag{2}$$

where $\otimes$ denotes the convolution computation, $j \in \{1,2,\ldots,7\}$ refers to the filter numbers of each color channel, e. g., $FX_{(blue,3)}$ is the result of the red channel by using the third filter of the blue group (in the bottom rectangle of Fig. 1).

For each color channel, there will be 7 resulting images produced by the corresponding filter group. Then we calculate the similarity between the filtering results of the reference and test images by

$$S_{(c,j)}(v,h) = \frac{2 \times FX_{(c,j)}(v,h) \times FY_{(c,j)}(v,h) + c}{\left(FX_{(c,j)}(v,h)\right)^2 + \left(FY_{(c,j)}(v,h)\right)^2 + c} \quad (3)$$

where $S_{(c,j)}$ is the similarity between $FX_{(c,j)}$ and $FY_{(c,j)}$(the results of the j-th filter in c color channel), v and h denote the vertical and horizontal position of the value in $S_{(c,j)}$, respectively, C = 190 is a constant.

After above computation, all the seven filtering results within each color channel will be combined by the following pooling scheme:

$$MAP_c = \sum_{j=1}^{7} S_{(c,j)} \quad (4)$$

where MAP_c is the similarity map between the reference and test images in each color channel. It represents the visual difference information within the tree color channels. Then we calculate the mean value of the MAP_c by

$$m_c = \frac{1}{V \times H} \sum_{v=1}^{V} \sum_{h=1}^{H} MAP_c(v,h) \quad (5)$$

where V and H denote the height and width of the similarity map (MAP_c), v and h denote the vertical and horizontal position of the value in MAP_c.

$$SD_c = \sqrt{\frac{1}{V \times H} \sum_{v=1}^{V} \sum_{h=1}^{H} \left(MAP_c(v,h) - m_c\right)^2} \quad (6)$$

Then the SD and the MS score can be computed by merging the values of the three color channels together.

$$SD = SD_{red} + SD_{green} + SD_{blue} \quad (7)$$

$$MS = m_{red} + m_{green} + m_{blue} \quad (8)$$

Finally, the PCD quality index is given by merging the two parts together.

$$PCD = SD - MS \times \alpha \quad (9)$$

where $\alpha = 0.2$ is a parameter to adjust the weight of MS.

IV. Experiments and Results

In this section, we will provide experimental results of PCD and comparisons with other state-of-the-art IQIs. In order to provide quantitative measures on the performance of the objective IQIs, we followed the performance evaluation procedures employed in the video quality experts group (VQEG) FR-TV testing.

The PCD index was compared with five state-of-the-art IQIs including: PSNR, SSIM[13], visual information fidelity (VIF) pixel version[4], IW-SSIM[11] and VSI[12]. All the IQIs were evaluated on the CSIQ database[7]. The CSIQ database consists of 30 original images distorted by 6 types of distortions at four to five levels. There are 866 distorted images contained in this database.

The performance metrics adopted in our experiment include the Spearman rank-order correlation coefficient (SROCC), the Kendall rank-order correlation coefficient (KROCC), the Pearson's linear correlation coefficient (PLCC) and the root mean squared error (RMSE). Generally, higher SROCC, KROCC and PLCC while lower RMSE values indicate a better performance. Before the computation of the PLCC and RMSE values, we need to apply a regression analysis by a nonlinear mapping between the objective and subjective scores. The nonlinear mapping chosen to fit the data is a five-parameter logistic function that is given by

$$Quality(x) = \beta_1 \left(\frac{1}{2} - \frac{1}{1 + exp\left(\beta_2 (x - \beta_3)\right)}\right) + \beta_4 x + \beta_5 \quad (10)$$

A. Performance on CSIQ Database

Table I shows the experimental results of the four performance metrics for all the six IQIs. From this table, we can see that the proposed PCD index performs the best among the 6 IQIs on the CSIQ database. Moreover, as a visual illustration of the relationship between subjective data and the IQI scores, the scatter plot of subjective scores versus objective scores of PCD is provided in Fig. 2(f), for comparison, the scatter plots of PSNR, SSIM, VIF-p, IW-SSIM and VSI are also provided in Fig. 2(a)~(b). Each star in the scatter plots represents a test image from the CSIQ database. We can see that in the scatter plot of PCD, all the stars are more compact, which indicates that the PCD index has a strong correlation with human evaluations.

TABLE I. Performance Comparison Among the 6 IQIs on CSIQ Database

IQI	SROCC	KROCC	PLCC	RMSE
PSNR	0.8057	0.6078	0.8000	0.1575
SSIM	0.8755	0.6900	0.8612	0.1334
VIF-p	0.9194	0.7532	0.9278	0.0980
IW-SSIM	0.9212	0.7522	0.9144	0.1063
VSI	0.9423	0.7857	0.9279	0.0979
PCD	**0.9466**	**0.7929**	**0.9536**	**0.0790**

TABLE II. KROCC Comparison Among 6 Different Distortion Types

IQI	AWGN	JPEG	J2K	APGN	GB	GCD
PSNR	0.9363	0.8879	0.9361	0.9338	0.9291	0.8623
SSIM	0.8974	0.9543	0.9605	0.8924	0.9608	0.7925
VIF-p	0.9575	**0.9703**	0.9671	0.9510	0.9744	0.9345
IW-SSIM	0.9380	0.9660	0.9682	0.9057	**0.9781**	**0.9540**
VSI	0.9637	0.9615	0.9692	**0.9636**	0.9679	0.9505
PCD	**0.9669**	0.9573	**0.9735**	0.9429	0.9724	0.9517

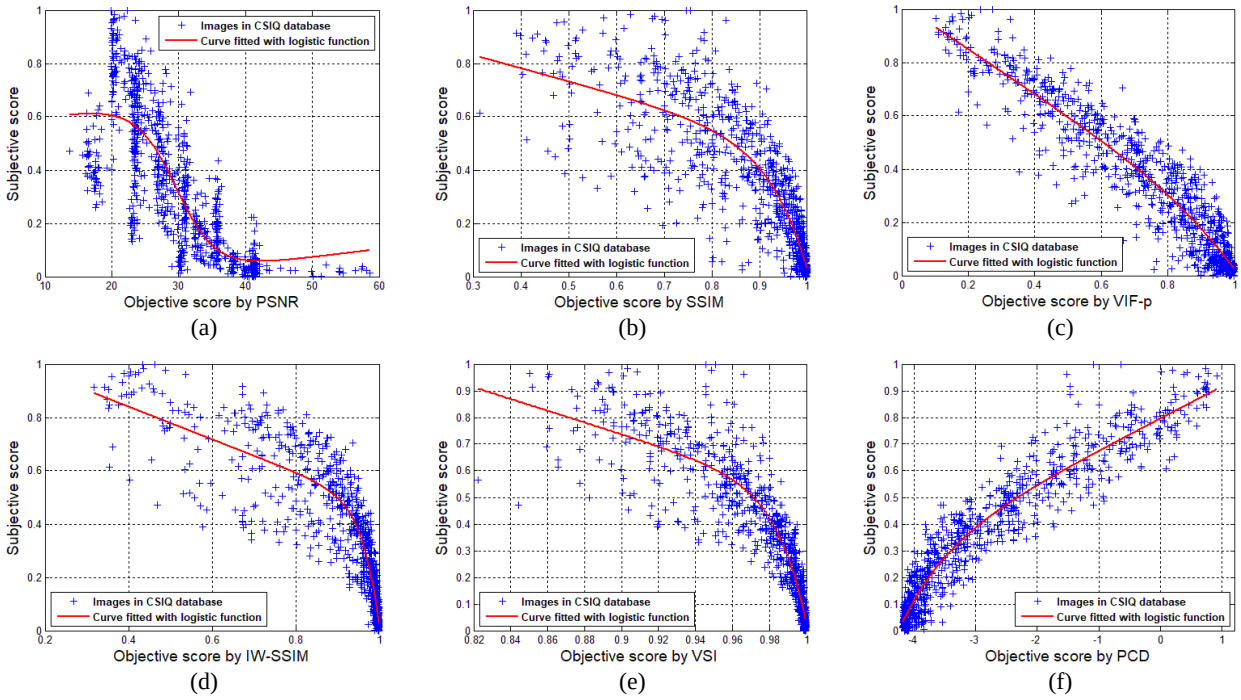


Figure 2. Scatter plots of the six IQIs on CSIQ database

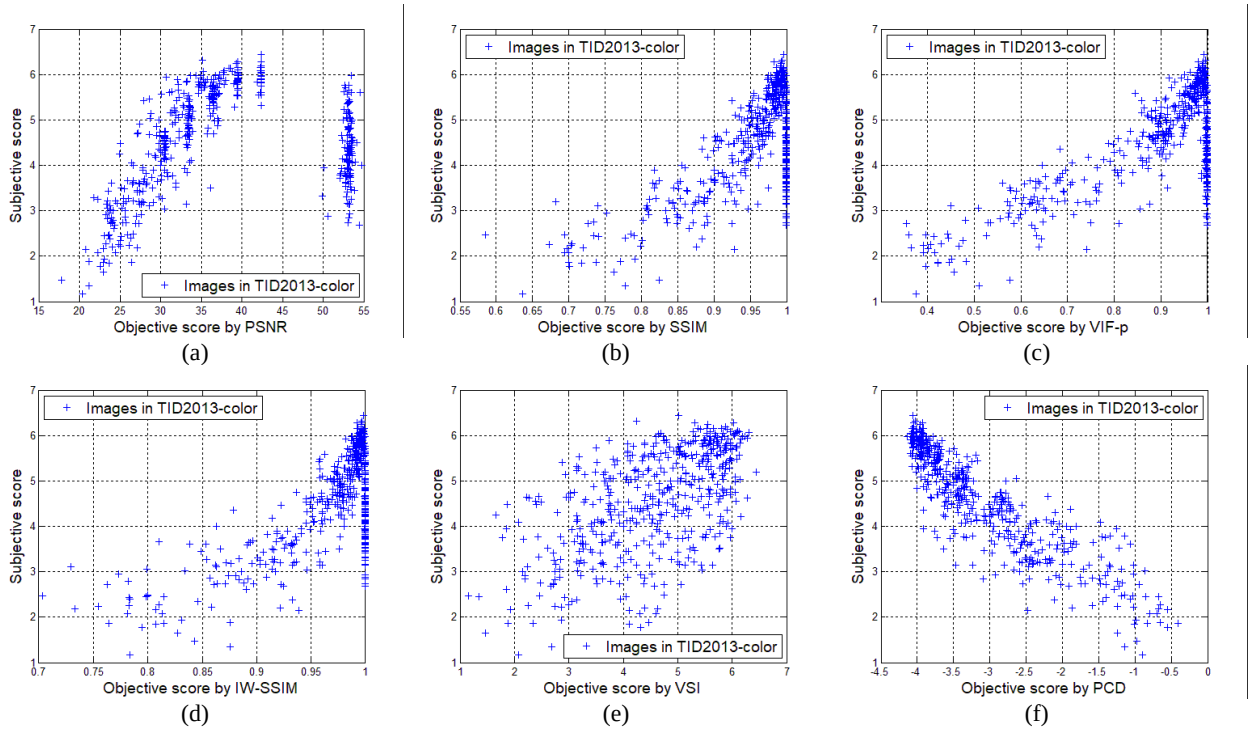


Figure 3. Scatter plots of the six IQIs on TID2013-color

The CSIQ has 6 distortion types: additive white Gaussian noise (AWGN), JPEG compression (JPEG), JPEG-2000 compression (J2K), additive pink Gaussian noise (APGN), Gaussian blurring (GB), and global contrast decrements (GCD). To completely evaluate each IQM index's ability to predict image quality degradations caused by specific types of distortions, we examined the performance of the competing methods on each distortion type. Table II shows the evaluation results on the 6 types of distortions. From Table II, we can see that PCD performs the best on AWGN and J2K distortion types. Some method can provide accurate results in some specific types of distortions, but performs

very bad on other distortion types, for example, IW-SSIM performs very well on GB and GCD distortion types but it performs not so well on APGN. From Table II, we can see that PCD's performance is very stable across the six distortions.

B. Performance on TID2013 Database

TID2013 database has the most distortion types[6]. There are 24 types of distortions for each reference image, and 5 levels for each type of distortion. We selected 4 distortion types that can lead to the distortion in color information and then formed a dataset namely TID2013-color. The 4 distortion types are as follows: additive noise in color components; quantization noise; JPEG compression; change of color saturation. The SROCC and KROCC results are listed in Table III. From Table III, we can see that PCD performs the best, which indicates that PCD can accurately predict the quality of images with color distortions.

The scatter plot of subjective scores versus objective scores of PCD is provided in Fig. 3(f), for comparison, the scatter plots of PSNR, SSIM, VIF-p, IW-SSIM and VSI are also provided in Fig. 3(a)~(b).

TABLE III. PERFORMANCE COMPARISON AMONG THE 6 IQIS ON TID2013-COLOR

IQI	SROCC	KROCC
PSNR	0.4049	0.3149
SSIM	0.3957	0.3029
VIF-p	0.4221	0.3384
IW-SSIM	0.3964	0.3005
VSI	0.5688	0.3957
PCD	**0.9028**	**0.7248**

C. Running time

We also measured the running time (in milliseconds) of each IQI on a 3.4-GHz CPU machine. All the results are listed in Table IV, from which we can see that PCD runs faster than VIF, IW-SSIM and VSI.

TABLE IV. COMPARISON OF COMPUTATION TIME IN SECONDS

IQIs	PSNR	SSIM	VIF-p	IW-SSIM	VSI	PCD
Time(ms)	2.17	12.89	287.33	131.62	113.09	53.60

V. CONCLUSIONS

A novel full reference IQI called principal component deviation (PCD) was proposed in this paper. The experimental results on the CSIQ and TID2013 image databases show that PCD performs better than the state-of-the-art IQIs (e. g., SSIM, VIF-p, and VSI) in terms of both accuracy and efficiency.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under Grant No. 61401404, No. 61502435 and No. 61602423, the Key Research Project of Colleges in Henan Province under Grant No. 15A520107, the Basic Research Project of Henan under Grant No. 142300410374, Science and Technology Project of Henan Province No. 192102210136.

REFERENCES

[1] A. C. Bovik. 2013. Automatic Prediction of Perceptual Image and Video Quality. *Proceedings of the IEEE* 101, 9 (September 2013), 2008–2024.

[2] Hua-Wen Chang, Hua Yang, Yong Gan, and Ming-Hui Wang. 2013. Sparse Feature Fidelity for Perceptual Image Quality Assessment. *Ieee Transactions on Image Processing* 22, 10 (October 2013), 4007–4018.

[3] Hua-wen Chang, Qiu-wen Zhang, Qing-gang Wu, and Yong Gan. 2015. Perceptual image quality assessment by independent feature detector. *Neurocomputing* 151, (March 2015), 1142–1152.

[4] H. R. Sheikh and A. C. Bovik. 2006. Image information and visual quality. *IEEE Transactions on Image Processing* 15, 2 (February 2006), 430–444.

[5] Aapo Hyvärinen, Jarmo Hurri, and Patrik O. Hoyer. *Natural image statistics: a probabilistic approach to early computational vision*. Springer.

[6] Nikolay Ponomarenko, Oleg Ieremeiev, Vladimir Lukin, Lina Jin, Karen Egiazarian, Jaakko Astola, Benoit Vozel, Kacem Chehdi, Marco Carli, Federica Battisti, and C.-C. Jay Kuo. 2013. A New Color Image Database TID2013: Innovations and Results. In *Advanced Concepts for Intelligent Vision Systems, Acivs 2013*, J. BlancTalon, A. Kasinski, W. Philips, D. Popescu and P. Scheunders (eds.). 402–413.

[7] Punit Singh and Damon M. Chandler. 2013. F-MAD: A Feature-Based Extension of the Most Apparent Distortion Algorithm for Image Quality Assessment. In *Image Quality and System Performance X*, P. D. Burns and S. Triantaphillidou (eds.). UNSP 86530I.

[8] M. E. Tipping and C. M. Bishop. 1999. Mixtures of probabilistic principal component analyzers. *Neural computation* 11, 2 (February 1999), 443–82.

[9] L. Wang and Q. Mao. 2019. Probabilistic Dimensionality Reduction via Structure Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 1 (2019), 205–219.

[10] Zhou Wang and Alan C. Bovik. 2009. Mean Squared Error: Love It or Leave It? A new look at signal fidelity measures. *Ieee Signal Processing Magazine* 26, 1 (January 2009), 98–117.

[11] Zhou Wang and Qiang Li. 2011. Information Content Weighting for Perceptual Image Quality Assessment. *Ieee Transactions on Image Processing* 20, 5 (May 2011), 1185–1198.

[12] Lin Zhang, Ying Shen, and Hongyu Li. 2014. VSI: A Visual Saliency-Induced Index for Perceptual Image Quality Assessment. *IEEE TRANSACTIONS ON IMAGE PROCESSING* 23, 10 (2014), 4270–4281.

[13] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (April 2004), 600–612.

Predicate Testing Generation for Safety-critical Systems

Wan Zhou
School of Computer and Information
Anhui Polytechnic University
Wuhu,China
2190720103@stu.ahpu.edu.cn

Yong Wang*
School of Computer and Information
Anhui Polytechnic University
Wuhu, P.R. China
State Key Laboratory for Novel
Software Technology, Nanjing University
Nanjing, 210000, China
yongwang@ahpu.edu.cn

Xiangyu Cheng Xue Wang
School of Computer and Information
Anhui Polytechnic University
Wuhu,China
c794452134@163.com

***Abstract*— *Safety-critical systems are those systems whose failure would cause significant property loss or crisis. These systems possibly contain complex predicates, which represent the conditions of system state transformation. Predicate testing is critical to ensuring system safety. However, it is difficult to generate appropriate test cases for complex predicates. To resolve the issue, predicate testing criteria based on major clause are proposed. Major clause of predicate is a clause that directly determines the value of predicate, and the preponderances of testing criteria based on major clause is that not only guarantee the rigor of testing, but also reduce the testing cost. This paper presents an approach that using different test criteria based on major clause to automatically generate corresponding test case constraint set for predicates in safety-critical systems. Firstly, the complex predicate is modeled as an abstract syntax tree (AST). Secondly, truth table for predicates is generated by extracting AST and further filtered based on predicate testing criteria (such as PC, RACC and CACC). Finally, testing case constraints set is generated by a random algorithm to satisfy certain test criteria. Our experimental results show that the constraint set generated by our method is reasonable and our approach is correct. By introducing variants into the empirical system, our approach shows the validity of different test criteria.*

Keywords- Safety-critical systems, predicate testing, major clause, predicate testing criteria

I. INTRODUCTION

Safety-critical systems refers to the systems which failure will cause significant property loss or crisis. It is diffusely used in automotive systems, power generation, distribution, avionics, medical systems and unmanned systems, etc. Considering the railway system as an example, the high degree of automation in the railway system is a significant factor affecting system safety, which involves the personal safety of passengers [2], [11], [15], [16].

Safety-critical systems possibly contain complex predicates. For instance, it contains more than three clauses, which represent the conditions of system state transformation. That is, predicates faults in safety-critical systems can cause software failures [4], therefore predicate testing is critical to ensuring system safety. Compared with common software systems, most safety-critical systems contain more logical judgment statements (i.e. predicates). Among these predicates, some are relatively brief, but there are also complex combinations of clauses [6]. Whether the predicates are simple or not, according to the strict security requirements of safety-critical systems, we must make reasonable and rigorous tests for the predicates.

However, it is difficult to generate appropriate test cases for complex predicates. Too much test cases may result in redundant test cases and increase test costs, and too few test cases make it difficult to guarantee the rigor of testing (i.e. more faults models can be found). To resolve the issue, predicate testing criteria based on major clause are proposed [9]. Major clause of predicate is a clause that directly determines the value of predicate, and the preponderances of testing criteria based on major clause is that not only guarantee the rigor of testing, but also reduce the testing cost. These testing criteria has been applied in the testing of aerospace systems. CACC, RACC and GACC are the strictly defined predicate test criteria based on major clauses [9].

This paper presents an approach using different test criteria based on major clause to automatically generate corresponding test case constraint set for predicates in safety-critical systems. The specific implementation process is as follows: Firstly, the complex predicate is modeled as an abstract syntax tree (AST). Secondly, truth table for predicates is generated by extracting AST and further filtered based on predicate testing criteria (such as PC, RACC and CACC). Finally, testing case constraints set is generated by a random algorithm to satisfy certain test criteria. Our experimental results show that the constraint set generated by our method is reasonable and our approach is correct. By introducing variants into the empirical system, our approach shows the validity of different test criteria.

The rest of this paper is organized as follows: Section2 explains the basic theoretical knowledge related to logical statements and predicate test criteria, providing a theoretical basis for subsequent testing. Section3 describes the overview and details of approach to realize the automatic generation of test case constraint sets for predicate coverage criteria based on major clauses. Section4 introduces the analysis process and experimental results of the empirical case safety-critical system Thermostat using our approach. Section5 discusses the shortcomings of our approach, and puts forward the future work. Section6 summarizes the whole paper.

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00032

II. PRELIMINARY

Before introducing the automated test methodology, we need to introduce some basic concepts about logical statements to predigest the subsequent testing. In different domains, these basic concepts may differ somewhat in nomenclature, but in general, the concepts they express are basically the same. In this paper, we use a common way in discrete mathematics to formalize logic expressions.

A. Notation and Terminology

- **predicate**: A *predicate* is an expression with a value of *true* or *false*. For example, the expression $(a \vee b)$ is a *predicate*. Predicates can be formed by combining Boolean clauses with Boolean operators and the operator has a priority. Table 1 shows the Boolean operators by operational priority [10]. It is worth mentioning that the starting point of this article is to consider logical expressions (i.e. from the point of view of predicates semantic) and not the internal structure of expressions. Which means the predicate $(a < b) \vee c \wedge d$ is treated no differently from $A \vee B \wedge C$.

TABLE I. THE BOOLEAN OPERATORS BY OPERATIONAL PRIORITY

$\neg$	The negation operator
$\wedge$	The and operator
$\vee$	The or operator
$\rightarrow$	The $implication$ operator
$\oplus$	The $exclusive$ operator
$\leftrightarrow$	The $equivalence$ operator

- **clause**: A *clause* is a predicate which does not contain any logical operators [10]. For instance, the predict $a > b \wedge (a < c)$ has two clauses $(a > b, a < c)$.

- **Predicate constraints**: Predicate constraints use the BR symbol to constrain Boolean variables or relational expressions [12]. The BR symbol generally contains $\{>, <, =, t, f\}$. In this article, the internal structure of the expression is ignored, which means predicates only be formed by combining Boolean clauses with Boolean operators, so the BR symbol is $\{t, f\}$ and the set of constraints for test cases consists of true or false.

- **Determination**, **major clause** and **minor cluse**: A *major clause* c_i for the predicate p, we say that the c_i *determines* p if the minor clauses $c_j \in p$, $j \neq i$ have value so that changing the truth value of c_i will change the truth value of p [1]. It's important to note that we're not saying that c_i has to have the same value as p. Consider the predicate p: $\neg a$, the value of a is never the same as the p, but the a determines the p.

- **AST** (*Abstract Syntax Tree*): In the process of executing the source code with the python language, python generally converts the source code into AST, and then converts AST to bytecode by traversing the AST of source code for compilation [5]. Boolean expressions can be expressed as AST as shown in Figure 1. Each leaf node of the AST (p) represents a Boolean variable or a relational expression, and the internal nodes of the AST (p) are Boolean operators such as ($\vee$,$\wedge$, $\neg$, $\oplus$). The purpose of using the AST in this article is to convert the source code into the AST during compilation so that leaf nodes can be extracted, which correspond to clauses in predicate.

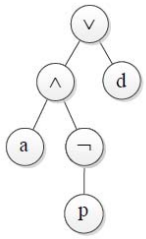


Figure 1. the AST of predicate $p: a \wedge \neg p \vee d$.

B. Logic Expression Coverage Criteria

The test of the predicate in this article is to override the logical expression (the predicate) with a set of test case constraints. The coverage criterion is constraint on the set of test case constraints. Concepts of predicates and clauses are used to introduce different coverage criteria. Now defining P is a set of predicates, and C is a set of clauses. For each $p \in P$, defining C_p are clauses in p, that is to say $C_p = \{c | c \in p\}$.

- CRITERION **Predicate Coverage (PC)** [1]: For each $p \in P$, TR (Test Requirement) has two requirements: p should evaluate to true and false.

- CRITERION **Correlated Active Clause Coverage (CACC)** [1]: For each $p \in P$ and each major clause $c_i \in C_p$, choose minor clause c_j, $i \neq j$ so that c_i determines p. TR has two requirements for each c_i: c_i should evaluate to true and false. And the values chosen for the c_j must cause p to be true for one value of the c_i and false for the other, that is, it is required that $p(c_i = true) \neq p(c_i = false)$.

- CRITERION **Restricted Active Clause Coverage (RACC)** [1]: For each $p \in P$ and each major clause $c_i \in C_p$, choose minor clause c_j, $i \neq j$ so that c_i determines p. TR has two requirements for each c_i: c_i should evaluate to true and false. The values chosen for c_j should be the same when c_i is false and when c_i is true. That is, it is required that $c_j(c_i = true) = c_j(c_i = false)$ for all c_j.

The coverage criteria PC also called DC (Decision Coverage). The PC criterion is sample, but it has deeper problems. To be specific, when we introduce a test for a clause, we also hope clauses impacting the value of the predicate, whereas the PC criteria concerned only with the value of the predicate [7]. Therefore, the PC criterion is only used in this paper for comparison with other criteria. There is also a predicate test criteria GACC (Restricted Active Clause Coverage) based on the major clause, it has a serious downside: Because it does not require the value of the predicate, it is unreasonable that we could satisfy the GACC without satisfying the predicate override. Therefore, GACC is not considered in this paper.

III. THE DETAILS AND OVERVIEW OF APPROACH

The approach realizing the automatic generation of test case constraint sets for predicates coverage criteria based on the major clause is roughly divided into the following steps, the details of each step will unfold in the following sections and the Figure 2 shows the framework of the approach:

1) **Step1**: Automatic generation of truth table of in-putting predicate.
2) **Step2**: Constraint set filtering based on the major clause: For each C_p as a major clause, filter out all possible cases of the test case constraint sets.
3) **Step3**: Generate test case constraint sets based on coverage criteria: For each C_p as a major clause, according to PC, RACC and CACC criteria, the test case constraint set generated in step 2 is further filtered and random combination constraint set generated.

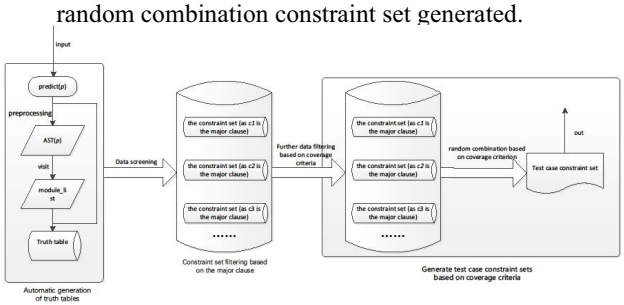


Figure 2. The framework of the approach.

A. Automatic generation of truth tables

The truth table is all the possible combinations of clauses values in a predicate. For a simple example, considering the predicate $p: a \vee b$, table 2 is the truth table for p.

TABLE II. THE TRUTH TABLE FOR $p: a \vee b$

a	b	p
1	1	1
1	0	1
0	1	1
0	0	0

Obviously, for a predicate with n clauses, there are 2^n possible combinations of clauses values in truth table. Because a predicate even in safety-critical system only have a finite number of clauses (generally not too many), this amount of computation is perfectly acceptable for computers, and necessary for predicate testing. Furthermore, python's syntax mechanism can automatically calculate the value of a predicate based on the value of clauses.

In order to implement automatic truth table generation. there are two critical questions:

Question1: How to automatically identify clauses in predicates? This paper uses AST to solve this problem skillfully. AST is an abstract representation of the syntactic structure of the source code. This article uses module *ast* to automatically convert the input predicate p into AST (p), and leaf nodes of AST (p) correspond to clauses in predicate p.

Question2: How to implement all combinations of clause values? We use the *product* method in the *itertools* module to generate cartesian products of multiple lists or iterators.

Then, we specifically implement the automatic generation of truth tables: Firstly, accepting the predicate p by input. Secondly, the predicate p is converted to an AST (p) by the parse method in the *ast* module. Visit the leaf nodes in the AST (p) and return the 'name' (i.e. id of nodes) of the nodes. Thirdly, we use the *PrettyTable*, passing two parameters (the predicate and the clause) into table. Add all the values of the clauses and the values of the corresponding predicate p to the table. At this point, the automatic generation of the predicate truth table is complete.

The pseudocode of algorithm 1 shows the appeal process. In the algorithm 1, the lines 1-6 solve the question1, and the lines 7-8 solves the question2.

Algorithm 1: Automatic generation of truth tables

```
   Input: predicate p
   Output: t - truth tbale of (p)
1  get_AST← ast.pares(p) ;                          /* AST← p */
2  function leaf_nodes(AST):
3     modules←visit_Module(AST);
4     visit_Name(Modules);
5     return Modules.id;
6  base ← leaf_nodes(get_AST);                  /* base↔clauses */
7  //generate the sets of booleans for the bases
8  conditions← list(itertools.prdouct([1,0], repeat=len(base)))
9  t ← PrettyTable(base +predicate p );          /* empty list */
10 for i ∈ conditions do
11    row ←i + predicate p;
12    t.add_row(row)
13 end
14 return t;
```

B. Constraint set filtering based on the major clause

In the previous section, we can get a truth table for any input predicate. The next step is to implement filtering based on the major clause test case constraint set. According to the definition of the major clause, we need to introduce a crucial conclusion, which provides a theoretical basis for filtering operations.

Conclusion: Considering predicate p and its truth table, for each clause c_p, when c_p as the major clause, there must

be two sets of test case constraints in the truth table that satisfy the following conditions: 1. The value of c_p are different in these two constraints sets. 2. All minor clauses have same values. Proof of the conclusion:

1) According to the definition of the major clause. When the minor clauses take specific values such that c_p as the major clause, and changing the value of c_p must cause the value of p to change.
2) Therefore, when c_p as the major clause, there must be situations in which the value of predicate p and major clause c_p is different and the value of the minor clause is the same.
3) Since the truth table contains all possible cases for clauses values, we can definitely find these two sets of test cases in the truth table.

Algorithm 2: Constraint set filtering based on the major clause

```
Input: predicate p
Output: truth_T truth_F
// The same operation with P_T and P_F, so it just shows P_T.
P_T ← [ ], truth_T ← [ ];
for i ∈ conditions do
    if row[-1] == 1 then
        P_T.append(row);
    end
end
new_P_T ← copy.deepcopy(P_T);     /* used to return indexes */
for i ∈ (0, len(base)) do
    for f ∈ (0, len(P_T)) do
        del new_P_T[f][i];
        del new_P_T[f][-1];
    end
end
t_indexes ← [ ];
for i ∈ new_P_T do
    for i ∈ new_P_F do
        t_indexes.append( new_P_T.index(x, 0) );
    end
end
for i ∈ t_indexes do
    truth_T.append(P_T[i])
end
return truth_T
```

Then we can implement the test case constraint set filtering based on the major clause, the specific implementation steps are as follows:

Step1: For predicate p and its truth table. By judging the value of the predicate p, divide the truth table into two lists: One list stores all possible constraint sets when $p = true$, and call it as *P-T*. The other list stores all possible constraint sets when $p = false$, call it as *P-F*.

Step2: We create two new lists, called *new-P-T* and *new-P-F*. Using the method deepcopy, copying the contents of *P-T* and *P-F* to *new-P-T* and *new- P-F*. When c_p as the major clause, we delete respectively all possible values of c_p and the values of the predicate p from table *new-P-T* and *new-P-F*.

Step3: Then we search for duplicate constraint sets in lists *new-P-T* and *new-P-F* and return their indexes. Find the corresponding set of constraints in *P-T* and *P-F* based on indexes. We know from the conclusion that these constraint sets must satisfy the condition of c_p as the major clause. Divide these constraint sets into two parts according to the values of p, which called *truth-T* and *truth-F*.

The pseudocode of algorithm 2 shows the appeal process. It's worth noting that, in the pseudocode of algorithm 2, part of variables is from algorithm 1. In the algorithm 2, the line 1-7 implements step1, the line 8-14 implements step2, the line 15-24 implements step3.

Considering predicate $p: a \wedge (b \vee c)$. Figure 3 shows the constraint set filtering process based on clause a as the major clause.

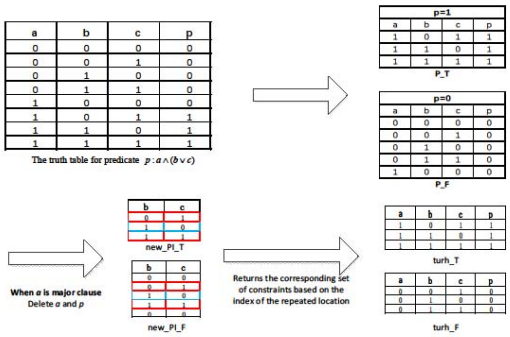


Figure 3. The process of constraint set filtering based on the major clause for predicate $p: a \wedge (b \vee c)$.

There is a method to check the correctness of algorithm 2. We need to find appropriate values for minor clauses in order to confirm major clause. There is a direct definitional approach that uses mathematical formula to find appropriate values for minor clauses.

For a predicate p, suppose each clause c occurs only once. Let $p_{c=true}$ represent the appearance of c every time in p replaced by true and $p_{c=false}$ represent the appearance of c every time in p replaced by $false$. Now we connect the $\oplus$:

$$p_c = p_{c=true} \oplus p_{c=false}$$

The p_c describes the accurate conditions under which c determines p. That is, if values of clauses in p_c are chosen, let p_c is true, then the truth value of c will determine the truth value of p; If values of clauses in pc are chosen, let p_c is false, then the false value of c will determine the false value of p.

For example, considering the predicate $p: a \wedge (b \vee c)$, when a is the major clause, we calculate p_a:

$$\begin{aligned} p_a &= p_{a=true} \oplus p_{a=false} \\ &= (true \wedge (b \vee c)) \oplus (false \wedge (b \vee c)) \\ &= (b \vee c) \oplus false \\ &= (b \vee c) \end{aligned}$$

So, a is the major clause only when $b \vee c$ is $true$ (i.e. at least one of b and c is true). In figure3, we can see that when a as the major clause, at least one of b and c is true. Similarly, we calculate p_b and p_c:

$p_b = a$, b is the major clause only when a is $true$.

$p_c = a$, c is the major clause only when a is $true$.

In figure3, we can see that when b or c as the major clause, a is true. The rationality of the experimental (for another predicate $p: a \wedge ((\neg A) \wedge (b \vee c))$) results was verified by the appeal mathematical formula. Therefore, the algorithm 2 is reasonable.

C. Generate test case constraint sets based on coverage criteria

We can further filter according to the constraint set obtained to meet the coverage criteria.

1) PC: The PC criterion only requires values of predicate. Therefore, we just need to respectively and randomly select a constraint set from P-T and P-F and combine them. The new constraint set will satisfy PC criterion.

2) CACC: The CACC criterion filters constraint sets based on major clauses: When c_p as the major clause, we get all the possible test case constraint sets truth-T and truth-F. we just need to respectively and randomly select a constraint set from truth-T and truth-F and combine them. The new constraint set will satisfy CACC criterion.

3) RACC: The RACC criterion filters constraint sets based on major clauses: When c_p as the major clause, we get all the possible test case constraint sets truth-T and truth-F. The indexes to the constraint set in truth-T and truth-F are corresponding because of the underlying compilation mechanism of the python language [14], which means when values of minor clauses are same, the location of constraint sets in truth-T and truth-F are same. So we traverse through all the elements in truth-T and return indexes, look for elements with same indexes in truth-F. Then using two new lists to store the constraint sets. We respectively and randomly select a constraint set from new lists and combine them. The new constraint set will satisfy RACC criterion.

The pseudocode of algorithm 3 implements PC, CACC and RACC criteria. It's worth noting that, in the pseudocode of algorithm 3, part of variables is from algorithm 1 and algorithm 2.

IV. THE EMPIRICAL CASE ANALYSIS

A. Preparatory work

1). **Experimental subject**: This experiment is oriented to the predicates in empirical case Safety-critical system Thermostat. The complete testing code of Thermostat has been uploaded to GitHub. Our experimental subject is the statement as show in figure4.
(https://github.com/zhouwan9/Thermostat)

```
if (((curTemp < dTemp - thresholdDiff) || (override && curTemp <
overTemp - thresholdDiff)) && (timeSinceLastRun > minLag))
```

Figure 4. Experimental subject.

2). **Experimental environment and tools**: The approach proposed in this paper is based on python3.7 and development platform is JetBrains PyCharm 2019.2.3 x64, the modules used include itertools, prettytable, re, ast, random and copy; The experimental subject's development language is Java; The development language for the test code is Java, JDK version is 1.8.0 221, and the test platform is Eclipse Java 2019-09.

3). **Mutation Testing**: The experimental subject of this paper is a complex logic judgment statement in Thermostat and the purpose of testing is to find Bugs in the program. In order to fully verify the correctness of the test results, mutation testing was introduced for comparison. Mutation testing is a fault-based software testing technique. The purpose of mutation testing is to find valid test cases and find real bugs in the program [17].

Mutation testing generates variants p' mutation operator to see if p' the same as the result of the original statement p, thus finding bugs.

Algorithm 3: Generate test case constraint sets based on coverage criteria

Input: *predicate p*
Output: The set of test case constraints that satisfy PC,CACC and RACC criteria

```
// Same operation with _T and _F,so it just shows _T.
PC ← [ ], CACC ← [ ], RACC ← [ ];
PC_T ← [ ];              /* Store the constraint sets as p = 1 */
// The PC criterion
for i ∈ base do
    if row[-1] == 1 then
        PC_T.append(i);
    end
end
random_PC_T ← [ ] ;
random_PC_T ← random.sample(PC_T, 1);
PC ← random_PC_T + random_PC_F;
// The CACC criterion
random_CACC_T += random.sample(truth_T, 1);
CACC ← random_CACC_T + random_CACC_F;
// The RACC criterion
indexes ← random.randint(0, len(truth_T) - 1);
// The set of constraints with the same index satisfies RACC
RACC_T ← [ ];
RACC_T.append(truth_T[random_index]);
RACC ← RACC_T + RACC_F;
return PC, CACC, RACC
```

B. Experimental process and results

The first step in the experiment is to formalize the subject. Based on the semantics of predicates, the internal structure of the expression is ignored in this article. That means for experimental subject is treated as:

Predicate $p_0: (a \vee (b \wedge c)) \wedge d$

a = curTemp < dTemp - thresholdDiff

b = override

c = overTemp – thresholdDiff

d = timeSinceLastRun > minLag

For clauses that ignore internal structure, typically, the only bugs at the program statement level are logical operator faults, and mutation operator '()' faults. Therefore, introducing the mutation operator LCR (Logical operator substitution) to get different variants ($p_1 - p_{10}$) for the predicate p_0. Strictly, p_4 is special, its mutation operator is AOR (Arithmetic operator substitution), that is '()' faults.

- p_1: $(a \vee (b \wedge c)) \wedge d$
- p_2: $(a \wedge (b \wedge c)) \wedge d$
- p_3: $\big(a \vee (b \wedge c)\big) \vee d$
- p_4: $a \vee (b \wedge c) \wedge d$
- p_5: $(\neg a \vee (b \wedge c)) \wedge d$
- p_6: $(a \vee (\neg b \wedge c)) \wedge d$
- p_7: $(a \vee (b \wedge c)) \wedge \neg d$
- p_8: $\big(a \wedge (b \vee c)\big) \wedge d$
- p_9: $\big(a \vee (b \vee c)\big) \vee d$
- p_{10}: $(a \wedge (b \wedge c)) \vee d$

The second step is to design the test code, using Junit unit tests. Unit tests are generally used to test the core functions of a project, it allows separate checks of properties in a program in a completely separate environment, also helping to find bugs in the development process [13]. Because PC, RACC and CACC criteria use random combinations of constraint sets, the resulting set of test case constraints is not necessarily same each time the test case constraint set which automatically generated. Therefore, we use five result sets of the test case constraint sets for each test criteria. Respectively calling them as:

- $T_{PC} = \{T_{PC1}, T_{PC2}, \dots, T_{PC5}\}$
- $T_{CACC} = \{T_{CACC1}, T_{CACC2}, \dots, T_{CACC5}\}$
- $T_{RACC} = \{T_{RACC1}, T_{RACC2}, \dots, T_{RACC5}\}$

		p_0	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
T_{PC}	T_{PC1}	0	50%	0	50%	0	50%	50%	50%	0	50%	50%
	T_{PC2}	0	50%	50%	50%	0	100%	50%	50%	0	50%	50%
	T_{PC3}	0	0	50%	50%	50%	50%	0	100%	0	50%	0
	T_{PC4}	0	0	50%	50%	50%	50%	0	100%	0	50%	0
	T_{PC5}	0	0	0	50%	50%	0	0	100%	0	50%	0
T_{CACC}	T_{CACC1}	0	33.3%	50%	50%	16.7%	66.7%	50%	66.7%	33.3%	50%	33.3%
	T_{CACC2}	0	33.3%	33.3%	50%	0	50%	50%	66.7%	16.7%	50%	33.3%
	T_{CACC3}	0	33.3%	50%	50%	16.7%	66.7%	50%	66.7%	16.7%	50%	33.3%
	T_{CACC4}	0	33.3%	50%	50%	16.7%	66.7%	50%	66.7%	33.3%	50%	50%
	T_{CACC5}	0	33.3%	50%	50%	16.7%	66.7%	50%	66.7%	16.7%	50%	50%
T_{RACC}	T_{RACC1}	0	33.3%	33.3%	66.7%	0	66.7%	66.7%	50%	33.3%	66.7%	50%
	T_{RACC2}	0	28.6%	42.9%	57.1%	14.3%	71.4%	57.1%	57.1%	28.6%	57.1%	42.9%
	T_{RACC3}	0	33.3%	50%	50%	16.7%	66.7%	50%	66.7%	33.3%	50%	33.3%
	T_{RACC4}	0	33.3%	33.3%	66.7%	0	66.7%	66.7%	50%	33.3%	66.7%	50%
	T_{RACC5}	0	33.3%	33.3%	66.7%	0	66.7%	66.7%	50%	33.3%	66.7%	50%

The last step, for p_0, using the approach proposed in this paper to automatically generate test case constraint sets T for experimental objects and do corresponding unit tests. Then for $p_1 - p_{10}$, doing the corresponding unit tests according to T. Table 3 shows the results of our experiment, the percentage number represents the percentage of failed test case constraint sets in the total test case constraint set.

C. Experimental analysis and conclusions

Analysis of experimental results:

1). Since our set of test case constraints is generated by the predicate p_0, all tests on p_0 will pass.

2). From the table6, we can see PC criterion is precarious. Because, more than once, it couldn't detect the variants. For example, T_{PC1} couldn't detect the variants p_2, p_4 and p_8. Therefore, in safety-critical system, PC criterion should be eliminated in the predicate testing.

3). When the test case constraint set fails, the test result is inconsistent with the expected result, which means, the variant was detected. Theoretically, the larger the percentage of failed test cases in the total test cases, the stronger the ability to detect variants.

4). It's worth noting that, the percentage here is not the probability of detecting variation. Considering Boolean operator faults (not include p_4), both CACC and RACC could absolutely detect variants. We can compare the abilities to detect variants using the average of this percentage. Table 4 shows the comparison of CACC and RACC criteria with the abilities to detect variants. In a comprehensive perspective, CACC and RACC criteria ability to detect variation was almost the same for Boolean operator faults.

TABLE IV. THE COMPARISON OF CACC AND RACC

	p_1	p_2	p_3	p_5	p_6	p_7	p_8	p_9	p_{10}	AVG
Avg (T_{CACC})	33.3%	46.7%	50%	63.4%	50%	66.7%	23.3%	50%	40.0%	47.0%
Avg (T_{RACC})	32.4%	38.6%	61.4%	67.6%	61.4%	54.8%	32.4%	61.4%	45.2%	50.1%

Experimental conclusions: This paper presents an approach to realize the automatic generation of test case constraint sets for predicates in safety-critical systems, which bases on CC, CACC and RACC criteria and applied to the predicate test in the empirical safety-critical systems Thermostat. Using this approach to automatically generate the test case constraint sets for predicate p_0: $(a \vee (b \wedge c)) \wedge d$. By introducing multiple variants ($p_1 - p_{10}$, but not include p_4) based the mutation operator LCR (Logical operator substitution), the experimental data show that the test case constraint sets automatically generated by our approach can effectively detect the logical operator errors of predicates. In summary, the test case constraint sets generated automatically using our approach is valid and reasonable.

V. DISCUSSION

A. Arithmetic operator faults between clauses

In the experiment with predicate p_0: $(a \vee (b \wedge c)) \wedge d$, we introduced a particular variant p_4: $a \vee (b \wedge c) \wedge d$, the

mutation operator of p_4 is AOR (Arithmetic operator substitution). Strictly speaking, p_4 is an arithmetic oper-ator fault. For clauses that ignore the internal structure of clauses, the arithmetic operator only includes (), which means arithmetic operator faults can only be '()' faults. The faults about () are loss increase and dislocation. To explore whether our approach can effectively detect the ' ()' faults in the relationship operator between clauses, we use our approach automatically generating test case constraint sets for $p_0: (a \vee (b \wedge c)) \wedge d$, and introducing the mutation operator AOR to generate more different variants for predicate p_0, using the same indicator (i.e. the percentage of failed test case constraint sets in the total test case constraint set) as table6. Before the experiment, we analyze p0 firstly: $p_0: (a \vee (b \wedge c)) \wedge d \leftrightarrow (a \vee b \wedge c) \wedge d$, '()' faults are rare because of the operational priority. Then we can only get two variants: $p_1: (a \vee b) \wedge c \wedge d$ and $p_2: a \vee b \wedge c \wedge d$. Table 5 shows the test results for p_1 and p_2.

TABLE V. THE TEST RESULTS FOR p_1 AND p_2

		p_0	p_1	p_2
T_{CACC}	T_{CACC1}	0	16.7%	16.7%
	T_{CACC2}	0	0	0
	T_{CACC3}	0	16.7%	16.7%
	T_{CACC4}	0	16.7%	16.7%
	T_{CACC5}	0	16.7%	16.7%
T_{RACC}	T_{RACC1}	0	0	0
	T_{RACC2}	0	14.3%	14.3%
	T_{RACC3}	0	16.7%	16.7%
	T_{RACC4}	0	0	0
	T_{RACC5}	0	0	0

It can be seen from table 5, the test case constraint sets automatically generated using our approach based on the CACC criterion could detect the arithmetic operator '()' faults between clauses that ignore the internal structure of clauses. However, there are risks, after several experiments, the possibility of CACC failed to detect arithmetic operator '()' faults between clauses is about 20%.

B. Future Work

Our future work is divided into three parts:

1). For arithmetic operator '()' faults between clauses that ignore the internal structure of clauses, we are trying to introduce a new approach that is 100% guaranteed to detect this faults, and integrate it with the approach that we're proposing now to make sure it could detect the arithmetic operator '()' faults between clauses and the Boolean operator faults of predicates.

2). After using our approach to automatically generate test case constraint sets for predicates in Safety-critical Systems. We want to have a prioritizing process for the set of test case constraints, and we want to prioritize testing the set of test case constraints that failed, this involves the accessibility of the program.

3). Before using our approach to automatically generate test case constraint sets for predicates in Safety-critical Systems, we will consider the accessibility of the program. Accessibility is defined as follows: The location or locations in the program that contain the fault must be reached [8]. We will use the existing accessibility analysis methods to improve our approach, it will allow our approach to consider accessibility while generating the constraint set.

VI. CONCLUSION

Safety-critical systems generally contain a mass of complex combinations of logic predicates, which directly affects the state of the program in safety-critical systems. Although the appearance of predicate test criteria allows developers to design rigorous test cases with reference while reducing the cost of testing, there are few automated tests for predicates in safety-critical systems, and most developers still focus on manual testing. For this purpose, this paper presents an automated test approach. This approach uses the predicate coverage criteria based on the major clause, automatically generating test case constraint sets for predicates in safety-critical system. The steps of the approach are roughly as follows: Firstly, we implement the automatic generation of predicate truth tables for the input predicates. Secondly, the set of predicate constraints is then filtered based on the major clause and the requirements of the predicate test criteria. Lastly, combine randomly the set of constraints to satisfy the predicate test criteria. Our experimental subject is a predicate for the empirical safety-critical system Thermostat, and the constraint sets generated based on PC criterion and test results on the variants as the contrast experiment. Our experimental results show that the constraint set generated by our method is reasonable and our approach is correct. For future jobs, we plan to improve the approach to ensure that more types of faults could be found. At the same time, the approach will consider the accessibility of the programing and the priority of the test cases.

ACKNOWLEDGMENT

This work was supported by the Anhui Natural Science Foundation (No. 1908085MF183), the Anhui University Natural Science Fund Key Project (Nos. KJ2018A0116, KJ2016A252, and KJ2017A104), Projects 61976005 and 61772270 supported by the NSFC of China, the Safety-Critical Software Key Laboratory Research Program (No. NJ2018014), the Training Program for Young and Middle-aged Top Talents of Anhui Polytechnic University (No. 201812), the Open Research Fund of Anhui Key Laboratory of Detection Technology and Energy Saving Devices (Anhui Polytechnic University)(No.DTESD2020B03), and the State Key Laboratory for Novel Software Technology (Nanjing University) Research Program (No. KFKT2019B23).

REFERENCES

[1] Paul Ammann, A. Jefferson Offutt, and Hong Huang. Coverage criteria for logical expressions. In 14th International Symposium on Software Reliability Engineering (ISSRE 2003), 17-20 November 2003, Denver, CO, USA, pages 99–107. IEEE Computer Society, 2003.

[2] Lionel C. Briand and Yvan Labiche. Empirical studies of software testing techniques: challenges, practical strategies, and future research. ACM SIGSOFT Software Engineering Notes, 29(5):1–3, 2004.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] Yusong Chen, Jianguang Chen, Yang Gao, Dongfeng Chen, and Yuming Tang. Research on software failure analysis and quality management model. In 2018 IEEE International Conference on Software Quality, Reliability and Security Companion, QRS Companion 2018, Lisbon, Portugal, July 16-20, 2018, pages 94–99. IEEE, 2018.

[5] Venkatesh Choppella, Garima Ahuja, and Aditi Mavalankar. How does a program run? A visual model based on annotating abstract syntax trees. In International Conference on Learning and Teaching in Computing and Engineering, LaTICE 2016, Mumbai, India, March 31 - April 3, 2016, pages 38–42. IEEE Computer Society, 2016.

[6] Vinicius H. S. Durelli, Jeff Offutt, Nan Li, M´arcio Eduardo Delamaro, Jin Guo, Zengshu Shi, and Xinge Ai. What to expect of predicates: An empirical analysis of predicates in real world programs. J. Syst. Softw., 113:324–336, 2016.

[7] Chunrong Fang, Zhenyu Chen, and Baowen Xu. Comparing logic coverage criteria on test case prioritization. Sci. China Inf. Sci., 55(12):2826–2840, 2012.

[8] A. Yu. Gerasimov, Leonid V. Kruglov, M. K. Ermakov, and Sergey P. Vartanov. An approach to reachability determination for static analysis defects with the help of dynamic symbolic execution. Programming and Computer Software, 44(6):467–475, 2018.

[9] Gary Kaminski, Paul Ammann, and Jeff Offutt. Better predicate testing. In Antonia Bertolino, Howard Foster, and J. Jenny Li, editors, Proceedings of the 6th International Workshop on Automation of Software Test, AST 2011, Waikiki, Honolulu, HI, USA, May 23-24, 2011, pages 57–63. ACM, 2011.

[10] Gary Kaminski, Paul Ammann, and Jeff Offutt. Improving logic-based testing. J. Syst. Softw., 86(8):2002–2012, 2013.

[11] Pramod Kumar, Lalit Kumar Singh, and Chiranjeev Kumar. Software reliability analysis for safety-critical and control systems. Quality and Reliability Eng. Int., 36(1):340–353, 2020.

[12] Mingyan Teng and Guangtian Zhu. Interactive search over web scale RDF data using predicates as constraints. J. Intell. Inf. Syst., 44(3):381–395, 2015.

[13] Fadel Tour´e, Mourad Badri, and Luc Lamontagne. A metrics suite for junit test code: a multiple case study on open source software. J. Softw. Eng. Res. Dev., 2:14, 2014.

[14] Hongyu Zhai, Casey Casalnuovo, and Premkumar T. Devanbu. Test coverage in python programs. In Margaret-Anne D. Storey, Bram Adams, and Sonia Haiduc, editors, Proceedings of the 16th International Conference on Mining Software Repositories, MSR 2019, 26-27 May 2019, Montreal, Canada, pages 116–120. IEEE / ACM, 2019.

[15] Dan Jing Zhang, Jian Hui Jiang, and Lin Bo Chen. Fault behaviors analysis of embedded programs. Journal of Computer Applications, 33(1):243–249, 2013.

[16] Tiantian Wang, Baiji Li, and Gongpeng Zhang. Application of Panel Data Model to Economic Effects of High-Speed Railway. International Journal of Performability Engineering, 16(7): 1130–1138, 2020.

[17] Guido Wimmel and Jan J¨urjens. Specification-based test generation for security-critical systems using mutations. In Chris George and Huaikou Miao, editors, Formal Methods and Software Engineering, 4th International Conference on Formal Engineering Methods, ICFEM 2002 Shanghai, China, October 21-25, 2002, Proceedings, volume 2495 of Lecture Notes in Computer Science, pages 471–482. Springer, 2002.

A Novel Bayesian Algorithm for Reliability of Exponential Model under Zero Failure Environment

Haiping Ren
Teaching Department of Basic Subjects
Jiangxi University of Science and Technology
Nanchang, China
chinarhp@163.com

Fan Zhang
Teaching Department of Basic Subjects
Jiangxi University of Science and Technology
Nanchang, China
Chinazf1985@163.com

***Abstract*— With the development of science and technology, the quality and lifetime of products become more better, and in type-I life testing, zero-failure data often occurs, especially in some high reliability and small sampling tests. Therefore the problem of zero-failure data becomes a new research hot topic in the reliability engineering. The aim of this paper is to observe the reliability estimation of the exponential distribution using a novel Bayesian algorithm. The new algorithm constructs the prior distribution of the reliability parameter on the basis of the memoryless property of exponential distribution. Bayesian estimators are derived under symmetric entropy loss function. At last, a practical example shows that the effectiveness of the proposed estimators.**

Keywords-Bayesian estimation; reliability; symmetric entropy loss function; zero failure data

I. INTRODUCTION

Reliability analysis of zero-failure data is a hot issue in reliability analysis in recent years. There are many literatures on the problem of zero-failure data using Bayesian method. Some achievements have been made in the reliability study of various lifetime distributions under zero-failure environment. For example, Han [1] studied Bayesian estimation and E-Bayesian estimation of failure rate and reliability when the prior distribution was truncated Gamma distribution of failure rate. Zhang [2] introduced a novel reliability analysis method of normal distribution with zero-failure data, and the proposed method can evaluate the reliability of products with high confidence level according to time-invalidation data. Zhang et al. [3] proposed a method for correction of fatigue life scatter factor based on time truncated zero-failure data of Weibull distribution. They pointed out that the method could fully make use of the data information and improve the forecasting precision of safe fatigue life significantly. Xiao and Ren [4] studied the HB estimation for reliability of Binomial distribution based on zero-failure data with the assumption of negative logarithm gamma prior of reliability under symmetric entropy loss function.

Exponential distribution is an important life distribution in the reliability analysis of engineering science [5]. The research and applications of exponential distribution have penetrated into many fields, such as quality control, computer science. In case of zero-failure data, Xu and Chen [6] observed the Bayesian interval estimation of reliability and failure rate for exponential distribution using a modified Bayesian credible limit method. Yin et al. [7] studied the estimation for the exponential distribution using E-Bayesian method. Górny and Cramer [8] developed generalized Type-I and Type-II hybrid censoring schemes to the case of progressively Type-II censoring. They derived exact likelihood inference for exponential distribution using the spacings' based approach under generalized progressive hybrid censoring schemes. Nowak [9] studied the maximum likelihood estimation (MLE) of the mean of the exponential distribution based on grouped samples which is stochastically increasing, and he also proved that the assumption of monotonicity of the sequence of distances in Balakrishnan et al. [10] can be dropped. Barakat et al. [11] constructed two pivotal quantities for deriving the prediction intervals for future lifetimes of exponential product based on a random number of generalized order statistics with random sample size.

Consider the type-I censored life test, and denote the censored times as t_i (i=1,2,…,m), where $t_1 < t_2 < \ldots < t_m$ and the corresponding t_i sample size is n_i. Then (t_i, n_i), $i = 1,2,\cdots,m$ is the zero-failure data set. Suppose that the life of a product distributed with an exponential distribution with probability density function

$$f(t \mid \lambda) = \lambda \exp(-\lambda t), \quad t > 0 \tag{1}$$

Here $\lambda > 0$ and $R = \exp(-\lambda t)$ is often called the reliability of exponential distribution at time t.

There are also many scholars devoted in the statistical inference and application of exponential distribution based on zero-failure data. But most of the existing Bayesian reliability estimations for exponential distribution are mainly observed under the square error loss function. Until recently, some Bayesian estimation of failure rate and reliability are discussed under some asymmetric loss functions, such as LINEX loss, entropy loss [12-15]. Symmetric entropy loss function is also a very useful loss function in Bayesian statistical inferences [16, 17], and it has the following mathematical expression:

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00033

$$L(\hat{R},R)=\frac{\hat{R}}{R}+\frac{R}{\hat{R}}-2, \tag{2}$$

Here $\hat{R}$ is an estimator of parameter R.

In this article, we will develop a new Bayesian algorithm in constructing the prior distribution of the reliability parameter with the help of memoryless property of exponential distribution. Furthermore we will study Bayesian estimation of failure rate λ and reliability R for exponential distribution under the symmetric loss function in case of zero-failure data.

II. Preliminary Knowledge

This section will introduce some basic concepts and notations about type-I censored life test with zero-failure data.

Assume that some products' lifetime distributed with exponential distribution (1). In order to evaluate the reliability of these products, the following tests are carried out: n samples are taken for life test, and they are divided into k groups, and the number of samples in i th group is n_i, where $n_1+n_2\cdots+n_k=n$.

The following notations are used to depict the considered problems:

(i) If $t=0$, then the failure probability $p_0=P\{T<0\}=0$ (or approximate zero).

(ii)The notation $S_i=n_i+\cdots+n_k$ means that there is S_i samples that has not been invalidated at time t_i, that is, there is S_i samples whose life is longer than time t_i.

(iii)At any time t_i, the failure probability is

$$p_i=P\ \ T<t_i\ \ ,$$

then we have

$$p_0<p_1<\cdots<p_k.$$

(iv) Let $R_i=1-p_i, i=1,\cdots,k$ be the reliability of exponential distribution at time t_i.

Since t_j time, S_j samples have participated in the test, and none of them has failed in the whole test process, so its likelihood function of R_j can be obtained as

$$L(R_j)=R_j^{S_j}, j=1,2,\cdots,k. \tag{3}$$

Under the symmetric entropy loss function (2), for unknown parameter R_j, the Bayesian estimator of R_j, can be easily derived as

$$\hat{R}_j=\sqrt{\frac{E(R_j\mid S_j,0)}{E(R_j^{-1}\mid S_j,0)}}. \tag{4}$$

In the following discussion, we will discuss the Bayesian estimation of reliability of $R_j(j=1,2,\cdots,k)$ of exponential distribution (1), where

$$R_j=1-p_j, j=1,\cdots,k. \tag{5}$$

III. Bayesian estimation of Reliability

A. Estimation of Reliability R_1

In this paper, the method introduced in Han [18] is adopted. In the case of zero failure data, the estimated failure probability p_1 is

$$\hat{p}_1=\frac{0.5}{S_1+1}, \tag{6}$$

and the obtained estimate R_1 is

$$\hat{R}_1=1-\hat{p}_1=\frac{S_1+0.5}{S_1+1} \tag{7}$$

B. Estimation of Reliability $R_j(j=2,3,\cdots,k)$

Because the distribution function $F(t)$ is a strictly convex function of t, and $F(0)=0$,then

$$\frac{F(t_1)-F(0)}{t_1-0}=\frac{p_1}{t_1}>\frac{F(t_2)-F(0)}{t_2-0}=\frac{p_2}{t_2}, \tag{8}$$

By assuming $p_1<p_2$, we can get

$$p_1<p_2<p_2',p_2'=\min(1,p_1\frac{t_2}{t_1}). \tag{9}$$

In Bayesian inference, how to determine a suitable prior distribution of the unknown parameter is especially important. But it is still a difficult problem especially in zero-failure data case, because there is little useful prior knowledge can be used in the estimation of an unknown parameter. To solve this problem, Han [18] developed the decreasing function methods as an alternative for the construction of prior probability density function of the failure probability p. The idea of decreasing method is: In the case of zero-failure data, a suitable kernel of the prior distribution of failure probability p should be in conformity with the larger possibility with respect to the smaller values of p and the smaller possibility with respect to larger values of p, then the prior probability density function of failure probability p should be a decreasing function. In the following discussion, we assume the kernel of prior density of failure probability p_2 is $1-p_2^2$, which is a decreasing function of p_2.

By an easily calculation, we can get the following lemma.

Lemma 1 Consider the type-I censored life test as mentioned in Section II, and denote the censored times as t_i (i=1, 2, …, m), where $t_1 < t_2 < \ldots, t_m$. Assume that the kernel of the prior density of failure probability p_2 is $1-p_2^2$. Then the prior distribution of $R_j (j>2)$ is

$$\pi(R_j) = \frac{3}{j-1}\left(\frac{R_j}{\hat{R}_1}\right)^{\frac{1}{j-1}-1} \cdot \frac{3(1-(1-\hat{R}_1\cdot\left(\frac{R_j}{\hat{R}_1}\right)^{\frac{1}{j-1}})^2)\hat{R}_1}{3((1-R)-(1-\hat{R}_1))-((1-R)^3-(1-\hat{R}_1)^3)},$$

where

$$\hat{R}_1(\frac{R}{\hat{R}_1})^{j-1} < R(t) < \hat{R}_1,$$

$$R = 1-\hat{p}_2' \ , \ \hat{p}_2' = \min(1, \hat{p}_1\frac{t_2}{t_1})$$

and

$$\hat{p}_1 = \frac{0.5}{S_1+1}.$$

Theorem 1. Consider the type-I censored life test as mentioned in Section II, and denote the censored times as t_i (i=1, 2, …, m), where $t_1 < t_2 < \ldots < t_m$. Assume that the kernel of the prior density of failure probability p_2 is $1-p_2^2$. Then under the symmetric entropy loss function (2), the Bayesian estimation of $R_j (j>2)$ is

$$\hat{R}_j = \hat{R}_1 \cdot \left(\frac{C(S_j+1)}{C(S_j-1)}\right)^{1/2}, \tag{10}$$

where

$$\hat{R}_1 = \frac{S_1+0.5}{S_1+1},$$

$$R = 1-\hat{p}_2' \ ,$$

$$\hat{p}_2' = \min(1, \hat{p}_1\frac{t_2}{t_1}),$$

$$\hat{p}_1 = \frac{0.5}{S_1+1}$$

and

$$C(x) = \hat{R}_1^2\{[\frac{2}{2/(j-1)+x}u^{\frac{2}{j-1}+x}]_{(R/\hat{R}_1)^{j-1}}^1 - [\frac{\hat{R}_1}{3/(j-1)+x}u^{\frac{3}{j-1}+x}]_{(R/\hat{R}_1)^{j-1}}^1\}. \tag{11}$$

Proof. Starting from time t_j, there are S_j samples participated in the test, and no failure sample occurred in the whole test process, so the likelihood function of R_j is

$$L(R_j) = R_j^{S_j}, j = 1,2,\cdots,k$$

By Lemma 1 and Bayes theorem, we can get the posterior density function of R_j as follows:

$$\pi(R_j \mid S_j, 0) = \frac{\pi(R_j)L(S_j,0;R_j)}{\int_{\hat{R}_1(R/\hat{R}_1)^{j-1}}^{\hat{R}_1}\pi(R_j)L(S_j,0;R_j)dR_j} = \frac{\left(\frac{R_j}{\hat{R}_1}\right)^{\frac{1}{j-1}+S_j-1}\cdot[1-(1-\hat{R}_1\cdot\left(\frac{R_j}{\hat{R}_1}\right)^{\frac{1}{j-1}})^2]}{C(S_j)}$$

Here $\hat{R}_1(\frac{R}{\hat{R}_1})^{j-1} < R(t) < \hat{R}_1$, $C(x)$ is defined in (11).

Then

$$E[R_j \mid S_j, 0] = \int_{\hat{R}_1(R/\hat{R}_1)^{j-1}}^{\hat{R}_1} R_j \cdot \frac{\pi(R_j)L(S_j,0;R_j)}{\int_{\hat{R}_1(R/\hat{R}_1)^{j-1}}^{\hat{R}_1}\pi(R_j)L(S_j,0;R_j)dR_j}dR_j$$

$$= \frac{\int_{\hat{R}_1(R/\hat{R}_1)^{j-1}}^{\hat{R}_1} R_j\cdot\left(\frac{R_j}{\hat{R}_1}\right)^{\frac{1}{j-1}+S_j-1}\cdot[1-(1-\hat{R}_1\cdot\left(\frac{R_j}{\hat{R}_1}\right)^{\frac{1}{j-1}})^2]dR_j}{C(S_j)},$$

$$= \frac{\hat{R}_1\cdot\int_{\hat{R}_1(R/\hat{R}_1)^{j-1}}^{\hat{R}_1}\left(\frac{R_j}{\hat{R}_1}\right)^{\frac{1}{j-1}+S_j+1-1}\cdot[1-(1-\hat{R}_1\cdot\left(\frac{R_j}{\hat{R}_1}\right)^{\frac{1}{j-1}})^2]dR_j}{C(S_j)}$$

$$= \frac{\hat{R}_1\cdot C(S_j+1)}{C(S_j)}$$

$$E[R_j^{-1} \mid S_j, 0] = \int_{\hat{R}_1(R/\hat{R}_1)^{j-1}}^{\hat{R}_1} R_j^{-1}\cdot\frac{\pi(R_j)L(S_j,0;R_j)}{\int_{\hat{R}_1(R/\hat{R}_1)^{j-1}}^{\hat{R}_1}\pi(R_j)L(S_j,0;R_j)dR_j}dR_j$$

$$= \frac{\int_{\hat{R}_1(R/\hat{R}_1)^{j-1}}^{\hat{R}_1} R_j^{-1}\cdot\left(\frac{R_j}{\hat{R}_1}\right)^{\frac{1}{j-1}+S_j-1}\cdot[1-(1-\hat{R}_1\cdot\left(\frac{R_j}{\hat{R}_1}\right)^{\frac{1}{j-1}})^2]dR_j}{C(S_j)}$$

$$= \frac{\hat{R}_1^{-1}\cdot\int_{\hat{R}_1(R/\hat{R}_1)^{j-1}}^{\hat{R}_1}\left(\frac{R_j}{\hat{R}_1}\right)^{\frac{1}{j-1}+S_j-1-1}\cdot[1-(1-\hat{R}_1\cdot\left(\frac{R_j}{\hat{R}_1}\right)^{\frac{1}{j-1}})^2]dR_j}{C(S_j)}$$

$$= \frac{\hat{R}_1^{-1}\cdot C(S_j-1)}{C(S_j)}$$

Then under the symmetric entropy loss function, Bayesian estimator of R_j is

$$\hat{R}_j = \left(\frac{E[R_j \mid S_j, 0]}{E[R_j^{-1} \mid S_j, 0]} \right)^{1/2} = \hat{R}_1 \cdot \sqrt{\frac{C(S_j + 1)}{C(S_j - 1)}} .$$

IV. AN ILLUSTRATED EXAMPLE

In a certain type of engine reliability test, no failure data obtained [19] as the first 4 columns of Table I. $t_2 - t_1 = t_3 - t_2 = \cdots = t_{13} - t_{12} = t = 112.51$, the unit of test is seconds, and a total of 13 groups of 51 data, relevant engineers believe that this type of engine has no failure after a large number of test results, so it is considered that its reliability is quite high, especially in the life of 1000 seconds, and the reliability will not be less than 0.95. Suppose that the life T of this type of engine obeys exponential distribution (1). Now our task is to give the estimate of reliability of exponential distribution at time $t_i, i = 1,2,\cdots,13$.

Now, using our proposed Bayesian algorithm as shown in Theorem 1, we can get the Bayesian estimates of $R_i, i = 1,2,\cdots,13$. The results are listed in Table I.

TABLE I. BAYES ESTIMATES OF $R_i, i = 1,2,\cdots,13$

i	1	2	3	4	5	6	7	8	9	10	11	12	13
t_i	100.2	212.7	325.2	437.7	550.2	662.7	775.2	887.7	1000.3	1112.8	1225.3	1337.8	1450.3
S_i	51	48	27	25	24	21	13	12	11	7	4	3	2
$\hat{R}_i$	0.9904	0.9862	0.9822	0.9785	0.9751	0.9716	0.9669	0.9633	0.9597	0.9545	0.9491	0.9445	0.9398

V. CONCLUSIONS

Because the reliability of products is getting higher and higher with the development of science and technology, the phenomenon of "no failure data" is more and more in reliability life test. In this paper, according to the memoryless property of exponential distribution, a prior distribution of reliability is constructed, and then Bayes estimation of average life is obtained based on zero-failure data.

In the future, we will study the Bayesian and E-Bayesian estimation for exponential distribution when the testing data is zero-failure data under other loss functions, such as Q-symmetric loss and scaled squared error loss functions.

ACKNOWLEDGMENT

The authors thank to the support of National Natural Science Foundation of China (No.71661012). Key Projects of Science Research of Jiangxi Province Educational Committee (No. GJJ170496).

REFERENCES

[1] M. Han, "The E-Bayesian estimation of the failure rate derived from exponential distribution in the case of zero-failure data," Mathematics in Practice and Theory, vol. 45, no. 5, pp. 172-178, March 2015.

[2] H. M. Fu, Y. B. Zhang, "Method of reliability analysis for time truncated zero-failure data based on normal distribution," Journal of Aerospace Power, vol. 25, no. 2, pp. 384-387, February 2010.

[3] Y. B. Zhang, H. M. Fu, Z. H. Wang, "Fatigue life scatter factor analysis for time truncated zero-failure data based on Weibull distribution," Journal of Aerospace Power, vol. 27, no. 4, pp. 795-800, April 2012.

[4] S. X. Xiao, H. P. Ren, "Hierarchical Bayesian reliability analysis of Binomial distribution based on zero-failure data," International Journal of Performability Engineering, vol. 14, no. 9, pp. 2076-2082, September 2018.

[5] H. S. Migdadi, M. H. Almomani, M. O. Abu-Shawiesh, and O. Meqdadi, "Reliability Performance of Improved General Series-Parallel Systems in the Generalized Exponential Lifetime Model," International Journal of Performability Engineering, vol. 15, no. 6, pp. 1734–1743, 2019.

[6] T. Q. Xu, Y. P. Chen, "Two-sided M-Bayesian credible limits of reliability parameters in the case of zero-failure data for exponential distribution," Applied Mathematical Modelling, vol. 38, no. 9-10, pp. 2586-2600, May 2014.

[7] Y. C. Yin, H. Z. Huang, W. Peng, et al., "An E-Bayesian method for reliability analysis of exponentially distributed products with zero-failure data," Eksploatacja i Niezawodnosc-Maintenance and Reliability, vol. 18, no. 3, pp. 445-449, June 2016.

[8] J. L. Górny, E. Cramer, "Exact likelihood inference for exponential distributions under generalized progressive hybrid censoring schemes," Statistical Methodology, vol. 29, pp. 70-94, March 2016.

[9] P. B. Nowak, "The MLE of the mean of the exponential distribution based on grouped data is stochastically increasing," Statistics & Probability Letters, vol. 111, pp. 49-54, April 2016.

[10] N. Balakrishnan, C. Brain, J. Mi, "Stochastic order and MLE of the mean of the exponential distribution," Methodology and Computing in Applied Probability, vol. 4, no. 1, pp. 83-93, March 2002.

[11] H. M. Barakat, E. M. Nigm, M. E. El-Adll, et al., "Prediction of future generalized order statistics based on exponential distribution

with random sample size," Statistical Papers, vol. 59, no. 2, pp. 605-631, June 2018.

[12] L. C. Hwang, "Second order optimal approximation in a particular exponential family under asymmetric LINEX loss," Statistics & Probability Letters, vol. 137, pp. 283-291, June 2018.

[13] P. K. Singh, S. K. Sing, U. Singh, "Bayes estimator of inverse Gaussian parameters under general entropy loss function using lindley's approximation," Communications in Statistics-Simulation and Computation, vol. 37, no. 9, pp. 1750-1762, October 2008.

[14] M. Nasr Esfahani, N. Nematollahi, "Admissible and minimax estimators of a lower bounded scale parameter of a Gamma distribution under the entropy loss function," Journal of Digital Imaging, vol. 4, no. 2, pp. 91-103, March 2009.

[15] Y. J. Du, X. X. Sun, "Estimation of scale parameter of Normal distribution under q-symmetric entropy loss function," Journal of Jilin University, vol. 50, no. 4, pp. 1044-1056, January 2007.

[16] B. Xu, D. H. Wang, R. T. Wang, "Estimator of scale parameter in a subclass of the exponential family under symmetric entropy loss," Northeastern Mathematical Journal, vol. 24, no. 5, pp. 447-457, May 2008.

[17] B. Long, F. Wang, C. X. Xi, "Bayesian estimation of the shape parameter for Lomax distribution based on progressively type-II censored tests," Statistics & Decision, no. 8, pp. 15-18, April 2017.

[18] M. Han, "Use of structure methods of prior distribution in reliability zero failure data," Operations Research and Management Science, vol. 7, no. 4, pp. 26-29, April 1998.

[19] H. B. Zhao, Y. M. Cheng, "Statistical analysis of zero-failure data using the memoryless property of exponential distributions," Chinese Journal Applied Probability and Statistics, vol. 20, no. 1, pp. 59-65, February 2004.

Reliability on Deep Learning Models: A Comprehensive Observation

Yuhong Zhang
School of Artificial Intelligence and Big Data
Key Laboratory of Grain Information Processing and Control, Ministry of Education
Henan University of Technology, Zhengzhou, China
Email: zhangyuhong001@gmail.com

Chunjing Xiao
Henan Key Laboratory of Big Data Analysis and Processing
Henan University, Kaifeng, China
Email: chunjingxiao@gmail.com

***Abstract*—This paper provides a comprehensive observation to examine the reliability of deep learning (DL) models. First, we will briefly introduce the essential background and kernel techniques in deep learning, such as downsampling and nonlinear discontinuity. Each of them may have some relation to the reliability of DL models. Then we discuss the inherent structural flaws of deep learning and the risk of unreliability that can result from it. Subsequently, we discuss various ways of generating adversarial samples that affect the DL model's reliability and corresponding preventive measures. Finally, we complete this observation by identifying current challenges and future trends for research.**

***Index Terms*—Deep neural networks, downsampling, adversarial example; reliability of deep model.**

I. Introduction

The past few years have witnessed the advances in artificial intelligence (AI) at a dizzy pace. There is no doubt that deep learning (DL) models are the most potent and high-profile method in AI [1]. Like Google, Amazon, and Alibaba, some top technology companies have all devoted marquee research and invested heavily in this technology. Nowadays, DL-based products greatly penetrate people's work and life. However, it is because of its widespread use, and we should focus on DL models' reliability and security.

DL models are known for their power to self-generate intermediate representations [2]. Technically, DL utilizes a multistage way to learn data representations, discovering the underlying structure in big data sets using various back-propagation (BP) variants. These representations can be obtained by composing nonlinear but straightforward modules, which transform the representation at a low level into a higher, usually more abstract level [3]. Some statistical techniques, such as pooling or sub-sampling and nonlinear mapping, can be adopted to achieve this abstraction or enhance DL models' representation. However, these technical issues significantly improve the model's performance in terms of classification at the cost of involving several potential reliability risks [4].

Some scholars have proved that there is some intriguing weakness of deep neural network models in image classification [5]. Their findings suggest that, despite their high accuracy in terms of the classification task, deep neural network models are surprisingly vulnerable to adversarial attacks [6]. The input-output mapping, which deep neural network learns from training data, is relatively discontinuous. By applying a specific hardly visual perturbation, the network can misclassify images. As a consequence, it could cause to maximize the prediction error of the network. In other words, neural networks are highly vulnerable to attacks of *adversarial examples* [7, 8]. These adversarial examples can easily fool the state-of-the-art DL models to misclassify the adversarial inputs, whereas still classified correctly by a human observer[9]. Some studies have also demonstrated that universal and tiny perturbation vectors exist, leading to a high probability of natural images being misclassified [10]. For instance, as shown in Fig. 1, the original image is labeled as "whale". However, when it is added the perturbation generated by DeepFool [11], then the corresponding fake image is classified as "turtle".

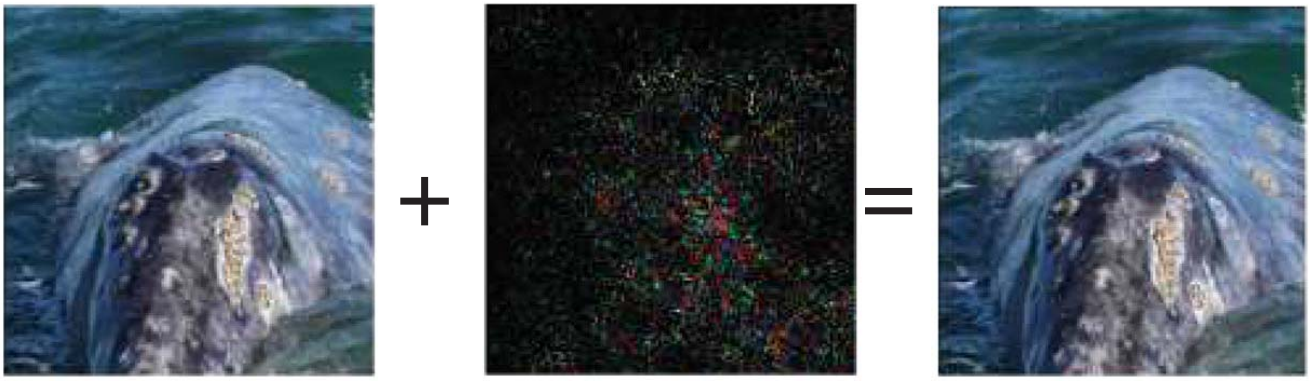
Adversarial examples pose reliability and security concerns since they could be used to attack the DL-based system [12]. It is, therefore, essential that DL models can resist attacks. Otherwise, they are as vulnerable as unprotected computers.

Unlike the review studies in related fields [13, 14], we observe the related studies from two aspects: internal structural defects of deep learning and external adversarial samples. We present in Section II the essence of a deep neural network, parts of which have inherent flaws, resulting in the possibility of being attacked. We then introduce more details on the unreliability caused by translational invariance in Section IV. In the subsequent Section V, it is dedicated to the different aspects of reliability about the DL models. In section VI, we discuss the challenges and possible directions that can work as guidelines for future research.

II. The Essence of Deep Neural Network

In this section, we briefly examine the fundamentals of deep learning. All kinds of artificial neural networks are composed of nodes and links, aiming to simulate neurons' behavior and synapses of a biological neural network. In a nutshell, deep neural networks try to fit a function from data that has a useful mapping between input and output. To accomplish this, the DL model requires the capability to abstract information layer by layer. Automatically learning non-local generalization from

978-0-7381-0497-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISSSR51244.2020.00034



(a) the original image (b) the minimal perturbation (c) the fake image

Fig. 1. An example of adversarial perturbations computed by DeepFool [11]

data is a strength of DL models. However, it also produces some counter-intuitive properties [5].

Deep Learning can be classified into two divisions, depending on whether it is trained in a supervised manner (by using labeled data) or an unsupervised manner (by using unlabeled data). For the supervised settings, a DL model is given an input χ and output, usually a class probability vector Y. More formally, the DL model can be regarded as a multidimensional function $F : (\chi; \omega) \mapsto Y$, where χ is a (raw) input vector, Y is an output vector, and ω is the connection weights between neurons in the neural network. A deep neural network can approximate nonlinear functions to arbitrary accuracy by regulating these weights [15, 16].

As is well known, a deep convolutional neural network (DCNN) is the most common and successful network structure in deep learning. Next, we mainly discuss some essential characteristics of DCNN, some of which can help us discover the potential defects that hit deep learning models' reliability. There are many variations of the DCNN architecture, but most of them are based on the layers' pattern.

The convolutional layer is considered as the core building block of DCNN. The convolutional layer uses a series of *learnable* filters (also called kernels) to identify the presence of specific features or patterns hidden in the original input. These filters are employed across the width and height of the input data in a sliding window manner. Then activation maps (also named feature maps) are calculated by a convolution operation, as shown in Fig. 2. Different filters can recognize different features. Subsequently, these activation maps are outputted, then passed to the next layer in the DCNN. The resultant output usually has a smaller (or the same) spatial dimension as the previous input.

Below we give a formal definition of the feature extractor. Let an image with three channels (namely, RGB), H in height and ω in width, be represented by $\chi \in \mathbb{R}^{H\times\omega\times 3}$. An L-layer DCNN can be regarded as a feature extractor $F_l(\chi) \in \mathbb{R}^{H_l\times\omega_l\times C_l}$, with layer $l \in \{0, 1, 2, \ldots, l\}$, spatial resolution $H_l \times \omega_l$ and C_l channels. Each feature map can also be up-sampled to original resolution, $\widetilde{F_l}(\chi) \in \mathbb{R}^{H\times\omega\times C_l}$.

III. The Components of Deep Learning Model with Potential Reliability Risks

A. Translational invariance by pooling

The pooling layer is usually inserted among successive convolutional layers. Pooling is also called the down-sampling. "*Sampling*" suggests that it helps progressively reduce the spatial size (width and height) of the previous layer's data over the network.

Pooling also has the additional effect of controlling overfitting and improving overall performance, while not losing essential information for feature extraction and classification.

There are some different kinds of pooling operations, like *max*-pooling (based on *max* operation), *average*-pooling (based on *mean* operation), as well as some advanced forms of pooling such as differentiable pooling and wavelet pooling.

The most frequently used operation of *down-sampling* is the *max* operation. For 2D max-pooling in the $H \times W$ image space, where each location (pixel) of the image corresponds to an input x_i, max-pooling after the feature encoding M_{x_i} can be applied to capture a larger spatial proximity of the features:

$$y = max\{|M_{x_1}, M_{x_2}, \cdots, M_{x_R}|\} \tag{1}$$

where M is the encoding matrix (different from each layer), $x_1, \cdots, x_R$ are the input vectors to the max-pooling operation.

With a 2×2 filter size, the *max* operation is taking the largest of four numbers in the filter area. Similarly, the average operation takes the mean of four numbers. The most typical setup for a pooling layer is applying 2^times2 filters with a stride of 2. This down-sampling operation will result in 75% of the pixel vector being discarded, as illustrated in Fig. 3.

Translation invariance is the result of pooling operations. Through pooling, the output of the convolutional network is the statistical result of a particular local region. In this case, even if we make some small changes to the input slightly,

the final result will not be affected. Taking the *max* pooling as an example, it is assumed that the pixel vector $\begin{pmatrix} 8 & 4 \\ 5 & 3 \end{pmatrix}$ inadvertently becomes $\begin{pmatrix} 5 & 4 \\ 8 & 3 \end{pmatrix}$, but its maximum still keep as (8).

Invariance to translation indicates that if we shift the inputs slightly, DCNN still identifies the class to which the input belongs. Therefore, translation invariance is a beneficial property when an object's exact location is not required. As a result, translational invariance is a much-needed property in many tasks, such as object recognition and audio recognition. For instance, we expect to detect a face by DCNN, even if the face slightly moves up or down (namely, locational translation), or the face is slightly rotated in the image (namely, rotational translation) [20].

A function $\widetilde{F}$ is translational equivariance if translating the input equally translates the output, which means that translation and feature extraction are commutable (Eq. 2).

$$T_{\Delta h,\Delta w}(\widetilde{F}((\chi)) = \widetilde{F}(T_{\Delta h,\Delta w}((\chi)) \quad \forall(\Delta h, \Delta w) \tag{2}$$

where T is the translation function. A representation is a translational invariance if translating the input results in an equal representation (Eq. 3).

$$\widetilde{F}(\chi) = \widetilde{F}(T_{\Delta h,\Delta w}((\chi)) \qquad \forall(\Delta h, \Delta w) \tag{3}$$

Many experiments have shown that max-pooling improves performance in terms of classification accuracy [21]. In some scenarios, however, this is taken for granted. Every coin has two sides. This feature also poses a risk to the deep model's reliability, which we will discuss later.

B. Activation layer as non-linear transformation

Activation functions play a significant role in the DL model. It is usually used to complete the neural network's nonlinear transformation to improve the whole network's representation capability. A neural network without activation function may "*degenerate*" into a linear regression model.

Most of the modern deep learning models use nonlinear activation functions. They allow models to generate complex mappings among the original inputs and the final outputs. The undivided linear sample space becomes linearly separable, through multi-layer nonlinear transformation, which, hopefully, makes the input data "nicer" for the network to classify, as illustrated in Fig. 4.

However, the deep learning model's reliability can not be guaranteed because of three significant factors [5]: The input-output mapping discontinuity caused by the deep neural network model's nonlinearity; The over-fitting caused by the insufficient model; Insufficient regularization. As a result, the linearity of high dimensional space is enough to cause the adversarial samples. The vulnerability of the depth learning model to adversarial samples is mainly due to its linear part. By transforming the model into a nonlinear radial basis function (RBF) network, the deep neural network model's vulnerability could be reduced [8].

IV. Unreliability Caused by Internal Structural Defects of Deep Learning

In the following sections, we discuss the deep neural network's reliability issue due to its structural weaknesses.

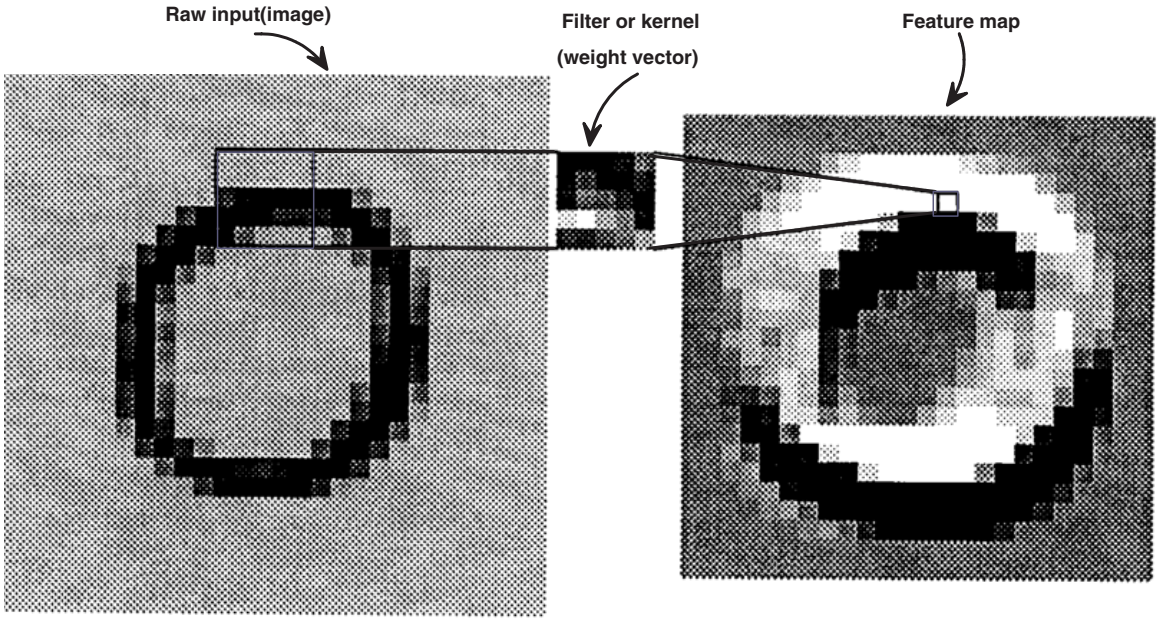


Fig. 2. A single convolutional filter on handwritten digit image in a convolutional network (reproduced from [17]).

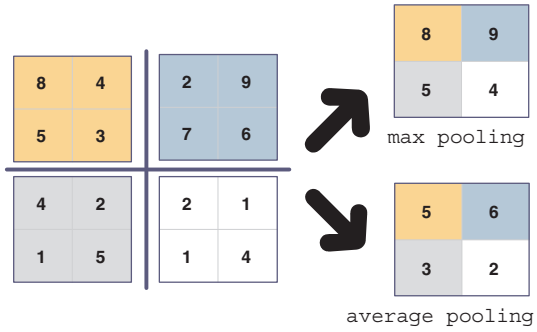


Fig. 3. A comparison of *max* pooling with *average* pooling for downsampling from the same receptive fields in DCNN

DCNNs are customarily assumed to be invariant to tiny image changes, leading to the classification task's robustness. However, max-pooling does not provide the anti-aliasing ability, and an unusual phenomenon recently occurred[23]. A small translation in the input may drastically change the classification result [24]. That is, very natural transformations, such as rotations or translations, can be used to fool the DL classification model completely [25]. In other words, defects in the training phase of DL models make them vulnerable to adversarial samples [4].

If an attacker finds a critical position in the model (that is, the critical position affecting the model's classification) and then only changes one pixel of this position, DCNN could misjudgment the classification. Many research endeavors are trying to figure out the critical position and fix the potential threat of the DCNN model [26].

The predicted correct class's probability will be significantly changed when rotating the image, as seen in Fig. 5. The baseline (black) exhibits chaotic behavior, which is upheld by a method called anti-aliased max-pooling (blue) [23].

As mentioned above, convolutions is translational equivariance, and pooling builds up translational invariance, however, striding takes depart from Nyquist–Shannon sampling theorem. As a result, it causes loss of shift-equivariance. Consider the example $[0,0,1,1,0,0,1,1]$ signal in Fig. 6 (subgraph (a)). Maxpooling (kernel $k = 2$, stride $s = 2$) will result in $[0,1,0,1]$ (subgraph (b), yellow curve). Simply shifting the input causes a much different result of $[1,1,1,1]$ (subgraph (c), yellow line). As a consequence, translation equivariance is lost, which may bring great hidden risk to the reliability of deep convolutional network model.

V. Different External Aspects of Reliability about Deep Neural Network Model

This section discusses the various external factors that influence the deep neural network model's reliability.

A. Adversarial examples in deep learning

In 2014, Szegedy et al. made an intriguing finding: even if state-of-the-art deep neural network models still are vulnerable to *adversarial examples* [5]. The input is formed by implementing small but intentionally perturbations to examples from the dataset. The perturbed input may result in a DL model outputting a wrong classification with high confidence [8]. Such adversarial examples, which they can be efficiently generated, boost essential security concerns about the reliability of deep neural network model [25].

Formally, for a given classifier, an adversarial perturbation can be defined as the minimal perturbation r which is sufficient to change the estimated label $\widehat{l}(\chi)$:

$$\Gamma(\chi;\widehat{l}) = \min_{r} \|r\|_2 \quad subject\ to\ \widehat{l}(\chi + r) \neq \widehat{l}(\chi) \tag{4}$$

where χ is an image data and $\widehat{l}(\chi)$ is the predicted label. $\Gamma(\chi;\widehat{l})$ is the robustness of $\widehat{l}$ at point χ. The robustness of the whole classifier $\widehat{l}(\chi)$ is then defined as

$$\rho_{adv}(\hat{l}) = \mathbb{E}_x \frac{\Gamma(\chi;\widehat{l})}{\|\chi\|_2} \tag{5}$$

where $\mathbb{E}_x$ is the expectation over the distribution of data.

Although adversarial attacks are classifier-specific, the adversarial perturbations seem to be generalizable across different DL models. So this is a security issue that deserves attention [11].

B. Different types of attacks

There are four primary attacks against the deep model: white box attack, black-box attack, targeted attack, and non-targeted target attack [27, 26]. Table I provides a summary of the main attributes of these attacks.

In a white box attack (also known as an open-box attack), an attacker can obtain all the model parameters and has an entire understanding of the training dataset. Although white-box attacks can easily fool DLs, it does not correspond to the real world. It is almost impracticable to obtain full information about the victim's DLs' structure and parameters [32].

On the contrary, a black box attack assumes little or no knowledge about the DL model and the training set. An attacker has no access to the model's parameters. The black box attack is possible because DL models have some shared vulnerability. Because neural networks simulate human intelligence, they may share similar semantic properties, although they differ in structures and weights. This kind of attack model is compatible with the actual situation, and this is the mainstream research on the DL model attack.

For a multi-classification network, the goal of a targeted attack is that enforce the DL model to get a specific class, which is different from the correct one. In contrast to a targeted attack, the non-targeted attack aims to fool the model to misclassify the adversarial sample. It does not have a particular goal. As long as the deep model's classification over the adversarial sample is wrong, the goal is accomplished.

From the perspective of implementation methods, adversarial attacks can be roughly categorized as gradient-based methods [33, 8], and optimization-based methods [34, 5]. Gradient-based methods search using the gradient direction,

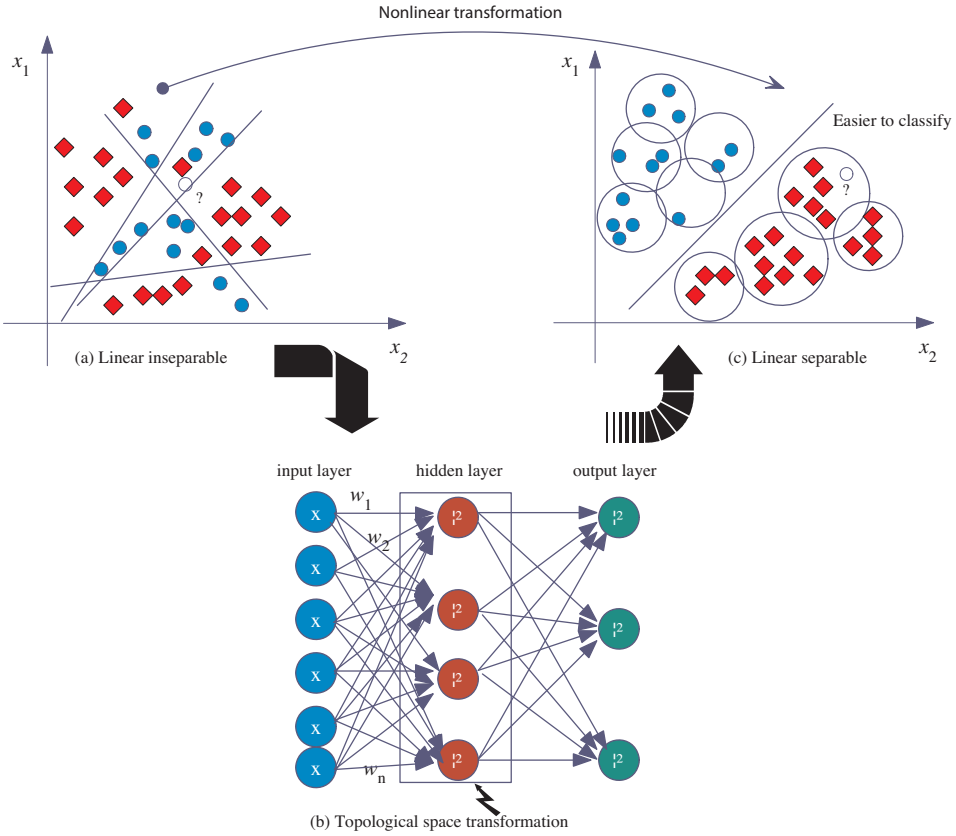


Fig. 4. The nonlinear transformation of the hidden layer makes classification easier

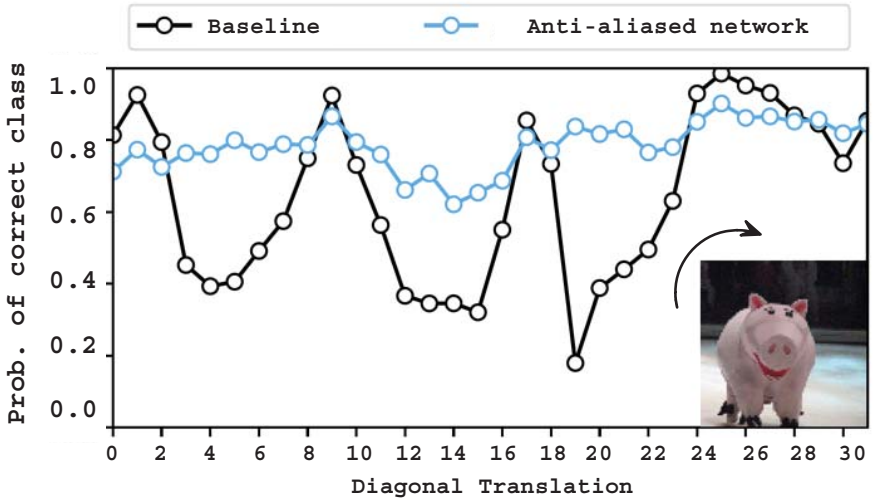


Fig. 5. Fluctuation of classification for selected images due to translational invariance (reproduced from [23])

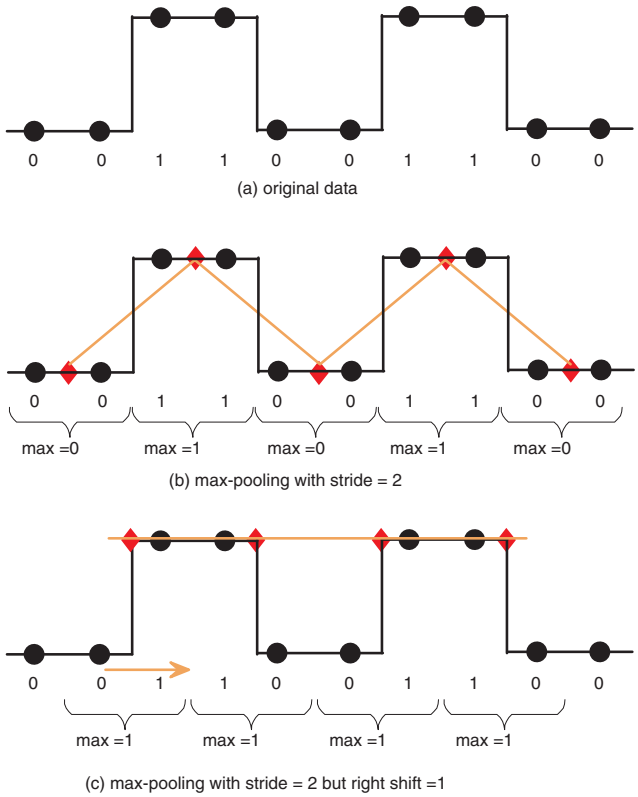


Fig. 6. Pooling would lose much of its functionality of shift-equivariance

Method	White/Black box	Targeted / Non-target	Image-specific / Universal	Perturbation norm	Learning type
L-BFGS [5]	White box	Targeted	Image-specific	l_∞	One-shot
FGSM [8]	White box	Targeted	Image-specific	l_∞	One-shot
I-FGSM [12]	White box	Targeted	Image-specific	l_∞	One-shot
JSMA [4]	White box	Targeted	Image-specific	l_0	Iterative
BIM & ILCM [12]	White box	Non-Targeted	Image-specific	l_∞	Iterative
One pixel [28]	Black box	Non-Targeted	Image-specific	l_0	Iterative
DeepFool [11]	White box	Non-Targeted	Image-specific	l_2, l_∞	Iterative
Universal perturbations [10]	White box	Non-Targeted	Universal	l_2, l_∞	Iterative
UPSET[29]	Black box	Targeted	Universal	l_∞	Iterative
ANGRI [29]	Black box	Targeted	Image-specific	l_∞	Iterative
Houdini [30]	Black box	Targeted	Image-specific	l_2, l_∞	Iterative
ATN [31]	Black box	Targeted	Image-specific	l_2, l_∞	Iterative
Attack on Attention [32]	White box	Non-Targeted	Universal	l_1	Iterative

TABLE I
THE ATTRIBUTES OF DIFFERENT ATTACKING METHODS

and the magnitude of perturbation is limited to avoid a significant distortion. Optimization-based methods regularly consider the magnitude restriction in the objective function. It generally considers the magnitude constraint in the objective function (also called cost function). In both cases, magnitude can be measured by l_1, l_2, l_∞-norm or other indicators.

C. Gradient-based attack approach

Many successful attacks are gradient-based methods, most of which belong to a white box attack. As early as 2014, Szegedy et al. first proposed the field of anti-attack and the concept of adversarial sample [5]. Szegedy et al. also proposed the BFGS (fast gradient sign method)-based attack model. This model computes adversarial samples by adding a pixel-wide perturbation of magnitude in the direction of the gradient. This perturbation is obtained with a single step. Thus it is very efficient in terms of computation time.

To resist adversarial perturbation, Gu & Rigazio explored network topology, pre-processing with denoising autoencoders (DAEs), and devised a model called Deep Contractive Network

(DCN) to enhance the robustness of DLs [35]. Nonetheless, applying DCN to large images is computationally expensive since the number of computations increases linearly with image pixels. Volodymyr et al. proposed a recurrent neural network (RNN) model, in which information can be deduced from an image or video by adaptively choosing a series of regions or locations and processing only selected regions [36].

To accelerate adversarial samples' calculation, the fast gradient sign method (FGSM) was proposed by Goodfellow et al [8]. By searching for the minimum perturbation of the loss function, neural network would mislabel the input, and the solving process by using FGSM can be transformed into a box-constrained optimization problem, which is defined as follows (Eq. 6):

$$\chi_{adv} = \chi + \varepsilon \cdot sign(\bigtriangledown_{\chi} J(\chi, y_{true})) \tag{6}$$

where χ is the input (clean) image, ε is a hyper-parameter to be chosen, χ_{adv} is the perturbed image, J is the cost function, and y_{true} is the label for the input χ.

Similarly to the FGSM, Kurakin et al. proposed a targeted fast gradient sign method (T-FGSM) [12]. In this method, a gradient step is computed by using the direction of the negative gradient with respect to the target category:

$$\chi_{adv} = \chi + \varepsilon \cdot sign(\bigtriangledown_{\chi} J(\chi, y_{target})) \tag{7}$$

where y_{target} is the label for the adversarial attack.

Compared to the iterative methods (I-FGSM) [12], the one-shot methods (both FGSM and T-FGSM) have lower success rate white box attacks. As a result, Kurakin et al. proposed a new construction method of adversarial sample with a more limited perturbation [12]. The iterative methods take T gradient steps of magiture$\alpha = \varepsilon/T$ instead of a single step t:

$$\begin{aligned} &\chi^{0}_{adv} = \chi, \\ &\chi^{t+1}_{adv} = \chi^{t}_{adv} + \alpha \cdot sign(\bigtriangledown_{\chi} J(\chi^{t}_{adv}, l_{true})) \end{aligned} \tag{8}$$

However, the iterative method (I-FGSM) leads to overfitting to a particular model. Papernot et al. [4] proposed constructing adversarial samples in the way of a white-box, called Jacobian saliency map attack (JSMA). This method's core idea is to produce adversarial samples by computing forward derivatives.

Moosavi-Dezfooli et al. introduced the DeepFool adversary, which selected which class an example is switched to. DeepFool was the first method to calculate the tiniest necessary perturbations and apply them against the adversarial samples, by using the l_2 norm to limit the perturbation range. The concluding result is better than FGSM and JSMA, but all three methods require considerable computing resources.

Subsequently, Moosavi-Dezfooli et al. [10] extended DeepFool and proposed a systematic algorithm for computing universal perturbations. Florian Tramer et al. [37] offered a method of adversarial training that remains vulnerable to black-box attacks, where perturbations are transferred to the undefended models. They used a small random step named *R + FGSM*. They proposed a simple but compelling single-step attack, called R+FGSM, which employs a small random perturbation to escape the data point's non-smooth vicinity. In 2017, N Carlini et al. introduced [34] a more efficient optimization problem, which could obtain more efficient adversarial samples by adding lower disturbance.

D. Differential evolution based method

An advanced case of the adversarial attack is to fool the classifier by changing only one pixel in the given image. The method termed "One Pixel Attack " was proposed by Su et al. [28].

(a) Cup (16.48%) **(b) Teapot (24.99%)**

Soup Bow (16.74) **Joystick (37.39)**

Fig. 7. One-pixel attacks on ImageNet where the modified pixel is highlighted with red circles

In this approach, differential evolution (DE) allows their method to generate an adversarial example without knowing network parameters or gradients. Two illustrative examples of the adversarial images from [28] are shown in Fig. 7. The initial class labels are in black. By contrast, the target class labels and their corresponding confidence are provided below.

For a clean image of I_c, they first produced a set of 400 vectors in $\mathbb{R}^5$ such that each vector contained the x and y coordinates and RGB values for any candidate pixel. They then randomly change the vector elements to generate offspring, who compete with their parents for fitness in the next iteration, using the network's probabilistic prediction labels as the fitness criteria. The last surviving child is utilized to adjust the pixels in the image. The only input that their technique requires is the probability labels of the target model's prediction. The DE-based method does not easily fall into the local minimum as the gradient-based method, nor does it need much knowledge about the target system to attack. Hence, the generated adversarial samples apply to various neural networks.

Another important finding in this research direction is that an adversarial patch, i.e., a small set of pixels instead of one pixel, can fool the deep network. Gao et al. [9] proposed a novel patch-wise attack method. Compared with the present perturbations at the pixel-wise level, their method has shown a more robust adversarial sample migration. It also can be easily coupled with existing state-of-the-art and implement a more sturdy attack.

E. Score-based method

Some attacks are more agnostic, relying only on confidence scores, such as class probabilities or logarithm, most of which usually belong to a black box attack. Papernot et al. [38] introduced one of the first black-box attacks against the deep neural network classifiers in cyberspace in the real-world settings, with no knowledge about the model. Using synthetic data, they practiced a network and instantiated attacks on Meta-Mind, Amazon, and Google remotely hosted neural networks. The results showed that the error classification of target networks was at rates of 84:24%, 96:19%, and 88:94%, respectively.

Chen et al. [39] also proposed an effective black-box attack, named 'Zeroth Order Optimization (ZOO)', which only had access to the input (images) and the output (confidence scores) of a targeted DL. ZOO directly estimated the gradients of the targeted DL for generating adversarial examples. This kind of attack was inspired by C&W attacks [34].

$$\begin{aligned} &minimize\ e_x \left\|x - x_0\right\|_2^2 + c \cdot f(x,t) \\ &\qquad subject\ to\ x \in [0,1]^p \end{aligned} \tag{9}$$

Eq. 9 is improved based on the original optimization. x_0 is the original image, x represents the changed image, t is the redirected category, $f(x,t)$ denotes x is classified as t's loss function (or confidence), and the anti-attack problem is transformed into the optimization one that minimizes the sum of the two parts. By exploiting the ZOO's method, the improved attacks to the targeted DL models can be completed by reducing the need for training substitute models and evading the loss in attack transferability.

In general, black-box adversarial attacks require much effort to attain successful adversarial examples visually indistinguishable from the original input. Most of the methods depend on substitute model training, gradient estimation, or genetic algorithms, often requiring too many queries. Thus, they may not be suitable for real-world systems where the maximum number of queries is limited due to their high cost. Ru et al. [40] introduced a query-efficient black-box attack that used Bayesian optimization to find successful anti-perturbations with high query efficiency.

L. Meunier et al. [41] made use of an evolutionary algorithm by borrowing ideas from l_∞ white-box attacks, in demand for derivative-free optimization. Du et al. [42] adopted meta-learning to approximate the gradient estimation. This method can significantly lessen the number of queries without affecting the attack success rate.

However, most of the existing adversarial attack model can only fool a black-box model at a low success rate. Addressing this problem, Dong et al. [43] proposed a broad class of momentum-based iterative algorithms to raise adversarial attacks.

F. Transfer-based method

The transfer-based method is a method between black-box and white-box attacks. This transfer-based approach attack relies on the similarity between the victim DL model and the attacked DL model, which serves as the surrogate model in a black-box attack.

Papernot et al. [38] introduced a way to attack a black-box network. They generated adversarial samples by using another accessible network, which performed the same task, and used these adversarial samples to attack the black-box network. This strategy proved to be reliable.

Although there have been some encouraging studies newly [44, 45], the transfer performance of models is not satisfying. A higher attack rate could be reached when two DL models have similar network structures, but it conflicts with the black-box attack's goal.

To clarify the trade-offs between accuracy and robustness, Su et al. [46] thoroughly benchmark 18 ImageNet models using multiple robustness metrics, including the success rate, distortion, and transferability of adversarial examples among 306 pairs of models.

G. Decision-based method

Most methods used to produce perturbations rely on detailed model knowledge (gradient-based attacks) or confidence scores, namely probability-like (score-based attacks). However, neither scenario exists in most real-world scenarios. Based on decision boundaries, Wieland Brendel et al. [47] offered a new decision-based attack method, which could find adversarial examples by only use the final decision. They also launched the first practical attack for general machine learning algorithms and complex natural data sets: the Boundary Attack.

The state-of-the-art adversarial approaches usually require time-consuming hyper-parameter tuning or multiple iterations to resolve adversarial base-based optimization. To fix this problem, Yao et al. [48] proposed a new family of trust region-based adversarial attacks to calculate adversarial perturbations efficiently. This method is useful for solving non-convex optimization problems.

VI. Discussions and Conclusions

The literature reviewed indicates that deep neural networks' reliability has attracted more and more attention due to inherent structural defects in deep learning. Therefore, adversarial attacks pose a physical threat to the deep neural networks model.

In the previous sections, we have conducted a comprehensive review of the recent literature on deep neural networks' reliability. Since some impressive facts have been discussed in those sections along with the technical details, we will make more general observations about this emerging research direction below.

- **More general scenario about deep neural network reliability**. Most current research has focused on fooling deep neural networks (mainly CNNs) with classification and recognition tasks. However, it is easy to observe, based on the investigated literature, the other neural model like MLPs, RNNs, and auto-encoder and reinforcement learning are also vulnerable to adversarial attacks

in general. In the future, researchers can expand from static image to natural language processing (NLP) and video-related fields.

- **More investigation of network vulnerability**. On the one hand, the deep neural network itself is not very interpretable. On the other hand, as for the reasons behind the subtle perturbations that can fool the deep neural network model, many works are inconsistent, and better studies are needed to explore the reasons that affect the reliability of the deep neural network model.
- **More efficient adversarial sample generation technique**. The current generation of adversarial samples is usually time-consuming and requires many computing resources. Therefore, it is necessary to investigate convex optimization techniques with better performance, such as using meta-learning to optimize network queries.
- **Anti-vulnerability research is promising.** Although various defense techniques exist to the Dl model's adversarial attack, and the literature also shows that the defense model can be successfully attacked again by designing counter-countermeasures. This observation requires that the new defenses need to provide an estimate of their robustness against possible counter-countermeasures. For example, Generative Adversarial Network (GAN) is expected to be combined with anti-attack research. It can simultaneously construct more effective adversarial samples and implement a defense strategy that makes the model more robust.

The existing literature shows that the current DL models have the hidden risk of reliability in structure and face the adversarial attack samples, making the deep learning model's reliability and security increasingly focused on research. However, due to the high activity of deep learning in this research direction, it can be hoped that DL models can show enough reliability against the adversarial attacks in the future and better provide power for artificial intelligence development.

VII. Acknowledgments

This paper is partly supported by the National Natural and Science Foundation of China (Grant Nos. 61705061, 61975053, U1904120).

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[2] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[3] Yoshua Bengio. "Deep learning of representations: Looking forward". In: *International Conference on Statistical Language and Speech Processing*. Springer. 2013, pp. 1–37.

[4] Nicolas Papernot et al. "The limitations of deep learning in adversarial settings". In: *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE. 2016, pp. 372–387.

[5] Christian Szegedy et al. "Intriguing properties of neural networks". In: *arXiv preprint arXiv:1312.6199* (2013).

[6] Nilesh Dalvi et al. "Adversarial classification". In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2004, pp. 99–108.

[7] Battista Biggio et al. "Evasion attacks against machine learning at test time". In: *Joint European conference on machine learning and knowledge discovery in databases*. Springer. 2013, pp. 387–402.

[8] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples". In: *arXiv preprint arXiv:1412.6572* (2014).

[9] Lianli Gao et al. "Patch-wise attack for fooling deep neural network". In: *arXiv preprint arXiv:2007.06765* (2020).

[10] Seyed-Mohsen Moosavi-Dezfooli et al. "Universal adversarial perturbations". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1765–1773.

[11] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. "Deepfool: a simple and accurate method to fool deep neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2574–2582.

[12] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. "Adversarial examples in the physical world". In: *arXiv preprint arXiv:1607.02533* (2016).

[13] Anirban Chakraborty et al. "Adversarial attacks and defences: A survey". In: *arXiv preprint arXiv:1810.00069* (2018).

[14] Bahram Lavi et al. "Survey on Reliable Deep Learning-Based Person Re-Identification Models: Are We There Yet?" In: *arXiv preprint arXiv:2005.00355* (2020).

[15] Ding-Xuan Zhou. "Universality of deep convolutional neural networks". In: *Applied and computational harmonic analysis* 48.2 (2020), pp. 787–794.

[16] Andreas Heinecke, Jinn Ho, and Wen-Liang Hwang. "Refinement and universal approximation via sparsely connected relu convolution nets". In: *IEEE Signal Processing Letters* 27 (2020), pp. 1175–1179.

[17] Yann Le Cun et al. "Handwritten zip code recognition with multilayer networks". In: *[1990] Proceedings. 10th International Conference on Pattern Recognition*. Vol. 2. IEEE. 1990, pp. 35–40.

[18] Matthew D Zeiler and Rob Fergus. "Differentiable pooling for hierarchical feature learning". In: *arXiv preprint arXiv:1207.0151* (2012).

[19] Travis Williams and Robert Li. "Wavelet pooling for convolutional neural networks". In: *International Conference on Learning Representations*. 2018.

[20] H-C Shin et al. “Organ detection using deep learning”. In: *Medical image recognition, segmentation and parsing*. Elsevier, 2016, pp. 123–153.

[21] Dominik Scherer, Andreas Müller, and Sven Behnke. “Evaluation of pooling operations in convolutional architectures for object recognition”. In: *International conference on artificial neural networks*. Springer. 2010, pp. 92–101.

[22] Yoshua Bengio. *Learning deep architectures for AI*. Now Publishers Inc, 2009.

[23] Richard Zhang. “Making convolutional networks shift-invariant again”. In: *arXiv preprint arXiv:1904.11486* (2019).

[24] Aharon Azulay and Yair Weiss. “Why do deep convolutional networks generalize so poorly to small image transformations?” In: *arXiv preprint arXiv:1805.12177* (2018).

[25] Logan Engstrom et al. “A rotation and a translation suffice: Fooling cnns with simple transformations”. In: *arXiv preprint arXiv:1712.02779* 1.2 (2017), p. 3.

[26] Naveed Akhtar and Ajmal Mian. “Threat of adversarial attacks on deep learning in computer vision: A survey”. In: *IEEE Access* 6 (2018), pp. 14410–14430.

[27] Kui Ren et al. “Adversarial attacks and defenses in deep learning”. In: *Engineering* (2020).

[28] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. “One pixel attack for fooling deep neural networks”. In: *IEEE Transactions on Evolutionary Computation* 23.5 (2019), pp. 828–841.

[29] Sayantan Sarkar et al. “UPSET and ANGRI: Breaking high performance image classifiers”. In: *arXiv preprint arXiv:1707.01159* (2017).

[30] Florian Tramèr et al. “The space of transferable adversarial examples”. In: *arXiv preprint arXiv:1704.03453* (2017).

[31] Shumeet Baluja and Ian Fischer. “Adversarial transformation networks: Learning to generate adversarial examples”. In: *arXiv preprint arXiv:1703.09387* (2017).

[32] Sizhe Chen et al. “Universal Adversarial Attack on Attention and the Resulting Dataset DAmageNet”. In: *arXiv preprint arXiv:2001.06325* (2020).

[33] Aleksander Madry et al. “Towards deep learning models resistant to adversarial attacks”. In: *arXiv preprint arXiv:1706.06083* (2017).

[34] Nicholas Carlini and David Wagner. “Towards evaluating the robustness of neural networks”. In: *2017 ieee symposium on security and privacy (sp)*. IEEE. 2017, pp. 39–57.

[35] Shixiang Gu and Luca Rigazio. “Towards deep neural network architectures robust to adversarial examples”. In: *arXiv preprint arXiv:1412.5068* (2014).

[36] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. “Recurrent models of visual attention”. In: *Advances in neural information processing systems*. 2014, pp. 2204–2212.

[37] Florian Tramèr et al. “Ensemble adversarial training: Attacks and defenses”. In: *arXiv preprint arXiv:1705.07204* (2017).

[38] Nicolas Papernot et al. “Practical black-box attacks against machine learning”. In: *Proceedings of the 2017 ACM on Asia conference on computer and communications security*. 2017, pp. 506–519.

[39] Pin-Yu Chen et al. “Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models”. In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. 2017, pp. 15–26.

[40] Binxin Ru et al. “Bayesopt adversarial attack”. In: *International Conference on Learning Representations*. 2019.

[41] Laurent Meunier, Jamal Atif, and Olivier Teytaud. “Yet another but more efficient black-box adversarial attack: tiling and evolution strategies”. In: *arXiv preprint arXiv:1910.02244* (2019).

[42] Jiawei Du et al. “Query-efficient meta attack to deep neural networks”. In: *arXiv preprint arXiv:1906.02398* (2019).

[43] Yinpeng Dong et al. “Boosting adversarial attacks with momentum”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 9185–9193.

[44] Cihang Xie et al. “Improving transferability of adversarial examples with input diversity”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2730–2739.

[45] Jiadong Lin et al. “Nesterov accelerated gradient and scale invariance for improving transferability of adversarial examples”. In: *8th International Conference on Learning Representations, ICLR*. 2020.

[46] Dong Su et al. “Is Robustness the Cost of Accuracy?–A Comprehensive Study on the Robustness of 18 Deep Image Classification Models”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 631–648.

[47] Wieland Brendel, Jonas Rauber, and Matthias Bethge. “Decision-based adversarial attacks: Reliable attacks against black-box machine learning models”. In: *The International Conference on Learning Representations (ICLR)*. 2018.

[48] Zhewei Yao et al. “Trust region based adversarial attack on neural networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 11350–11359.

Author Index