Computers, Materials & Continua CMC, vol.65, no.2, pp.1837-1854, 2020

Research on Data Extraction and Analysis of Software Defect in IoT Communication Software

Wenbin Bi[1], Fang Yu[2], Ning Cao[3], Wei Huo[3], Guangsheng Cao[4, *], Xiuli Han[5], Lili Sun[6] and Russell Higgs[7]

Abstract: Software defect feature selection has problems of feature space dimensionality reduction and large search space. This research proposes a defect prediction feature selection framework based on improved shuffled frog leaping algorithm (ISFLA).Using the two-level structure of the framework and the improved hybrid leapfrog algorithm's own advantages, the feature values are sorted, and some features with high correlation are selected to avoid other heuristic algorithms in the defect prediction that are easy to produce local The case where the convergence rate of the optimal or parameter optimization process is relatively slow. The framework improves generalization of predictions of unknown data samples and enhances the ability to search for features related to learning tasks. At the same time, this framework further reduces the dimension of the feature space. After the contrast simulation experiment with other common defect prediction methods, we used the actual test data set to verify the framework for multiple iterations on Internet of Things (IoT) system platform. The experimental results show that the software defect prediction feature selection framework based on ISFLA is very effective in defect prediction of IoT communication software. This framework can save the testing time of IoT communication software, effectively improve the performance of software defect prediction, and ensure the software quality.

Keywords: Improved shuffled frog leaping algorithm, defect prediction, feature selection framework, Internet of Things.

[1] School of Computer and Software, Dalian Neusoft University of Information, Dalian, 116023, China.

[2] School of Information Engineering, Qingdao Bin Hai University, Qingdao, 266555, China.

[3] School of Internet of Things and Software Technology, Wuxi Vocational College of Science and Technology, Wuxi, 214028, China.

[4] Public Teaching Department, Neuedu Software Talent Training School, Qingdao, 266000, China.

[5] Public Teaching Department, Qingdao Technical College, Qingdao, 266555, China.

[6] School of Information Engineering, Sanming University, Sanming, 365004, China.

[7] School of Mathematics and Statistics, University College Dublin, Dublin, Ireland.

[*] Corresponding Author: Guangsheng Cao. Email: gsc_ssh@163.com.

Received: 04 March 2020; Accepted: 02 May 2020.

1 Introduction

The Internet of Vehicles (IoV) is an open network system that connects people, cars and the external environments together [Li, Liu and Yang (2014)]. As a system network, it exchanges a large lot of information among people, cars and the external environments based on LAN, mobile Internet and the external wide area networks in accordance with special communication protocols [Kim and Kim (2019)]. The system collects and filters a large amount of data from vehicles, and the platform processes the data and build up an integrated network for intelligent traffic management, intelligent dynamic information services and intelligent vehicle control [Wang, Deng, Xu et al. (2019)]. Over recent years, with the rapid development of technologies of wireless communication, automatic control and sensors, the Internet of Vehicles has, as one of the concrete manifestations of the Internet of Things in the field of transportation, been rapidly developed and applied. The Internet of Vehicles is an important way to deliver intelligent transportation, and will be an important part of the future smart cities [Lin, Huang and Pasha (2014)].

The application of the Internet of Vehicles in intelligent transportation will change people's traditional driving habits and travel modes, showing extremely broad development prospects [Wei and Wang (2018)]. With the development of vehicle intelligence and driverless technology, the vehicles' ability to perceive the surrounding environment and the computing and planning capabilities of the vehicle terminals have increased significantly [Zhang, Deng, Liu et al. (2019)]. Information transmits between vehicles, vehicles and people, vehicles and roads. This provides new scientific ideas and technical means for the overall management and control of intelligent transportation [Liu and Liu (2013)]. At present, the technologies and service capabilities of Internet of Vehicles are continuously improved [Duan, Wei, Tian et al. (2019)]. With the help of all-round connections and information interactions among people, vehicles, roads and cloud platforms [Zhu and Ge (2017)], a large number of new product applications have been spawned. The new applications also bring new development requirements for the intelligent and connecting level of automobiles and transportations, driving the development of connected-car technologies and industries and promoting the digital and intelligent development of cities [Wang, Yao, Han et al. (2018)]. Through the Internet of Vehicles, and in the form of human-server-vehicle communication, users can implement intelligent management and control of vehicles, covering people-vehicle information exchanges, remote controls, remote diagnoses and other functions [Huang (2013)].

The Internet of Vehicles software has the characteristics of openness, large scale and complexity, while facing such challenges as diverse application types, rapid application scale changes, fast software iteration updates, and many uncertain trust threats. The mentioned above challenges also bring a series of problems to the testing of IoV software, including huge testing workload, high testing cost and unstable test quality. In recent years, many scholars in the field of software engineering have proposed software defect prediction methods to improve the quality of software testing. Purposed to predict the number and type of undetected defects, the software defect prediction first appeared in the 1970s, and today there are hundreds of related software prediction models. Typical software defect prediction methods corresponding to software defect prediction models include: linear discrimination (LDA), Boolean difference function (BDF), classification

regression tree (CART), support vector machine (SVM), and artificial neural network (ANN) [Wang, Wu and Li (2008)].

Feature selection of software measurement metadata is critical in software defect prediction. The validity of software defect prediction depends on not only the quality of the measurement metadata, but also the dimension of the measurement metadata. If too many negative features, such as irrelevant features and redundant features, exist in the measurement metadata (the features that are closely related to measurement metadata are referred to as positive features) and result in a high dimension of measurement metadata. That will weaken the software defect prediction, and affect the related performance of software defect prediction. Ant colony optimization algorithm (ACO) [Kabir, Shahjahan and Murase (2012)] as a heuristic search algorithm simulating ant population evolution, is also applied to software defect prediction. However, in the ant colony optimization algorithm there is a lack of direct communication between groups, which is easy to produce local optimization or slow convergence in the later stage of parameter optimization process. The Shuffled Frog Leaping Algorithm (SFLA) [Yu, Li, Yang et al. (2018)], as a meta-heuristic algorithm, focuses on the communications between frogs in each group and the exchanges of information between groups, with both local and global collaborative searches. The Shuffled Frog Leaping Algorithm has the advantages of strong search ability, simple algorithm, great flexibility and strong robustness.

This research proposes a defect prediction feature selection framework based on improved shuffled frog leaping algorithm (ISFLA). The framework improves generalization of predictions of unknown data samples, enhances the ability to search for features related to learning tasks. At the same time, this framework further reduces the dimension of the feature space. This framework bases on the improved SFLA algorithm, and divides into two parts: preliminary selection and optimization of the software complexity measurement. In the primary selection part, the preliminary preprocessing of the collected relevant historical data sets is mainly to deal with contradictory data, repeated data, invalid data and redundant data, and select a more reasonable test historical data set.

The optimization part deals with the attribute set of software defect complexity, including local optimization within the feature group and global information interaction. In the process of local search and evolution in feature subgroups, the worst individuals in the subgroups randomly approach to the best individuals in the subgroups. The differences of all individuals in the group gradually narrow, and result in the aggregation effect. In the global information interaction part, the ranking bases on fitness value, and the ranking with high fitness value (high feature correlation) comes first. At the same time, we use a threshold to select the final feature subset and complete defect prediction process.

2 Related work

2.1 Related concepts and architecture of internet of vehicles system

Related concepts of the Internet of vehicles system include the following:

2.1.1 Telematics

It is a central control operating system installed in the car, helping to establish an interactive

interface between the user and the car. Bluetooth phone, reversing image, map navigation and other functions are usually integrated. Some also include functions such as maintenance appointments, remote diagnosis, voice recognition, vehicle rescue and other functions.

2.1.2 On-Board Diagnostic System (OBD)

OBD is the abbreviation of on-board diagnostic system, which can monitor the operation of the electronic control system and other multiple functional modules of automobile engine. Once an abnormal situation occurs, the fault is identified through a proprietary algorithm, and the corresponding Diagnostic Trouble Code (DTC) is given and stored in the memory. Maintenance personnel can read out the DTC using the diagnostic instrument, which can quickly locate the fault and reduce the time of manual diagnosis. In addition, the OBD interface can also read other data during the operation of the vehicle, such as vehicle speed, mileage, and fuel consumption.

2.1.3 Controller Area Network (CAN)

CAN is one of the most widely used Fieldbus in the world. It is widely used in the automotive industry. It can effectively support distributed control or real-time control, and is a type of serial communication. The CAN bus performs data transmission in message units. The structure of the CAN bus is very simple. One CAN_High and one CAN_Low, respectively representing high level and low level, are simple and stable.

2.1.4 Head up Display (HUD)

The HUD head-up display uses the optical principle of reflection to project information on the glass. When the driver reads the information, he can see without having to lower his head, and the transparent glass will not cause any obstruction to the driver's vision and will not affect driving safety. A large amount of driving information, such as speed, gear, fuel consumption, and endurance mileage can be displayed on the HUD, which is very convenient.

2.1.5 Intelligent vehicle interconnection system

Seamlessly connect the user’s smart phone with the Vehicle Mounted System to realize the interconnection between “people, vehicle and mobile phone”, giving drivers a more convenient experience. Baidu CarNet represents domestic products, and foreign products include Android Auto and CarPlay.

2.1.6 Intelligent rearview mirror

Automobile intelligent rearview mirror, usually running embedded operating system, replacing traditional rearview mirror with video stream, providing rich software functions, generally with Bluetooth calls, road navigation, driving records, online radio, online music, reversing images, driving track, parking monitoring and other functions. Major domestic manufacturers include smart hardware vendors such as Mijia, 360, and operators such as China Mobile.

2.1.7 Voice technology

Currently, more mature voice technologies include Tencent Cloud Voice, iFLYTEK Voice, and Baidu Voice. A wide range of voice technologies used in the Internet of Vehicles include voice calling, voice navigation, etc. The direction of application in the Internet of Vehicles is voice control, which is a very good way of human-computer interaction in the vehicle environment. Directly calling related services can avoid the complicated operation of manually operating the screen or buttons, and improves security. It is the development direction of human-computer interaction, so that drivers and passengers can enjoy the ubiquitous information services. In-vehicle network communication technology and voice technology will become a key part of Intelligent Vehicles [Wu (2016)].

2.1.8 Radio Frequency Identification (RFID)

In the Internet of Vehicles, there is a problem of identification between vehicles, between vehicles and road facilities. RFID helps to realize the information exchange between vehicles. This information can use in practical scenarios such as logistics industry and automobile payment. The Internet of Vehicles uses RFID technology combined with existing network technology, database technology, middleware technology, etc., to build a huge Internet of Things composed of a large number of networked RFID terminals. After data transmission and information fusion, we can extract content related to traffic jam and driving safety. The main advantage of RFID applied to the Internet of Vehicles is to be able to identify multiple objects in high-speed operation with technology, and to facilitate the mutual transmission of information between various vehicles in the Internet of Vehicles [Ma (2016)].

2.1.9 Intelligent Transport System (ITS)

With the development of economy, the number of cars is increasing year by year, and the problem of traffic congestion is becoming increasingly serious. Intelligent transportation system is a safe and efficient management system formed by applying advanced technology of the Internet of Things to the field of transportation. Through comprehensive perception of people, vehicle, and road-related information, and through big data analysis, it can provide scientific decisions for path planning, improve transportation efficiency, reduce energy consumption, ease road congestion, and reduce environmental pollution, which is an advanced Traffic Integrated Management System.

2.1.10 T-BOX

T-Box, as the fulcrum of cloud-to-vehicle information interaction in the system, not only plays the role of vehicle ECU, but also bears the heavy responsibility of wireless communication module [Wang, Jiang, Gu et al. (2018)].

The network architecture of the Internet of Vehicles divide into three layers: perception layer, network layer, and application layer, as shown in Fig. 1.

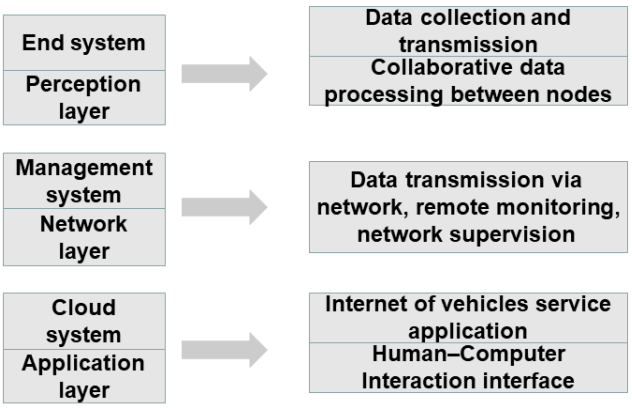

Figure 1: Network architecture of the Internet of Vehicles

Perception layer is the vehicle-mounted terminal device, it has the function of collecting vehicle-related information, including driving data and vehicle body status. It can also support remote control. The physical equipment mainly includes T-Box, vehicle electronic control unit, various sensors, etc.

Network layer mainly solves the interconnection of vehicles, roads, people, and cloud platforms. The communication between the vehicle's internal network and a variety of heterogeneous networks through mobile Internet devices and software is a bridge between the network in the vehicle and the external system.

Application layer mainly includes business applications and human-computer interaction interfaces [Ju, Zhao, Wen et al. (2018)], such as driving behavior analysis, vehicle trajectory planning, remote diagnosis, and scheduled maintenance.

2.2 Shuffled frog leaping algorithm

From a biological perspective, how can a group of frogs find food quickly and effectively? The methods of local optimization and information exchange among different subgroups are used to reference. First, a group of frogs divides into several subgroups. Second, the distance between each frog and food is used to communicate with other frogs. Compare the farthest distance with the nearest distance, and then move the corresponding position to adjust the distance. When optimizes the location of each subgroup to a certain extent, then the sub-groups in the total group are mixed to meet the final boundary conditions.

Eusuff proposed a new swarm algorithm, SFLA (Shuffled Frog Leaping Algorithm). The algorithm process of SFLA is as follows [Liu Wang, Huo et al. (2014)]: assuming the frog population size is Q, the initial population is $X = (X_1, X_2, \ldots, X_q)$. For the frog population generated by random method, the fitness function $f(x_i)$ of each frog individual is calculated, and the fitness value ranks in descending order according to the calculated fitness value. Then apply the method to the whole frog populations. The whole frog population was divided into E subgroups, each of which contained L frogs, namely T=E×L.

In the optimization process, X_1 is put into the first subgroup, X_2 into the second subgroup until the last individual into X_E .The frogs with the best fitness in each subgroup were recorded as G_b and the frogs with the worst fitness were recorded as G_w. The frogs with the best fitness in the whole group were recorded as X_b.

In each local search update strategy, only G_w is operated to improve its fitness. The internal search strategy of frog subgroup is as follows:

$$sd_{(s)} = R(G_b - G_w) \tag{1}$$

$$G_n = G_w + sd_{(s)}, -sd_{(\max)} \le sd_{(s)} \le sd_{(\max)} \tag{2}$$

R denotes the random number in [0, 1], and $sd_{(\max)}$ denotes the maximum step of frog movement.

The optimization rule of SFLA is that if the fitness of new frog individual G_n is greater than that of original frog individual G_w, then G_n will replace G_w to become a new individual in frog subgroup. Otherwise, replace G_b with X_b and re-operate according to Eqs. (1) and (2). If there is improvement, replace G_w; if there is no improvement, randomly generate an individual to replace G_w. Repeat the above until the frog population reaches the set number of iterations. After the local search of all frog subgroups, reorder all the individuals in the population and divide into subgroups. Then search the local parts of the subgroup and execute many times until the group evolution algebra.

2.3 Improved shuffled frog leaping algorithm

The new position of the worst frog may always be within the linear range between the current position and the optimal frog position. Even if we update the position of the worst frog according to the random solution, we cannot ensure that the random solution will be better. At the same time, the worst frog position updates only according to the optimal frog position, and the communication with other frog individuals is lacking, which reduces the adaptability of the frog population. In the hybrid leaping frog algorithm, the moving step of the worst frog is uncertain, and the learning step may be too large to omit the global optimum. In the later stage of optimization using SFLA algorithm, the difference between Frog individuals in the population decreases and the moving step decreases, which reduces the convergence accuracy of the whole algorithm.

In order to make the worst frog individuals fully communicate with other frogs, we import other frog individuals in the subgroup into the moving step of the improved SFLA algorithm. At the same time, through the adjustment of trigonometric function [Chang and Zhao (2016)], the improved SFLA algorithm performs better in both local search and global search. In addition, in order to improve the search accuracy, the newly generated frog individuals search within the maximum radius between themselves and the optimal frog in their subgroup with their own origin.

$$sd_{(k)} = 2 \times \sin(\frac{t\pi}{2G}) \times R \times [x(k)_b - x(k)_w] + R \times [x(k)_i - x(k)_w] \quad (3)$$

$$sd_{(k)} = 2 \times \sin(\frac{t\pi}{2G}) \times R \times [x(g)_b - x(k)_w] + R \times [x(k)_i - x(k)_w] \quad (4)$$

$$x_{new} = 2 \times \cos(\frac{t\pi}{2G}) \times R \times r_{\max} \times x(k)_s \quad (5)$$

where t=1, 2 ... G, G are fixed evolutionary iterations, $r_{\max}$ is the maximum radius between the frog individual and the optimal frog in the subgroup, and R is the random number between (0, 1). $sd_{(k)}$, is the worst moving step of the Kth frog population. $x(k)_s$, is a randomly selected frog individual from the Kth frog subgroup. $x(k)_b$, $x(k)_w$ and $x(g)_b$ represent the best, worst in the Kth frog population and globally best frog individuals. x_{new}, is a random new frog individual in frog sub-population.

3 Establishment of feature selection framework for software defect prediction based on ISFLA

Now we assume that there are $m \times n$ feature subsets.

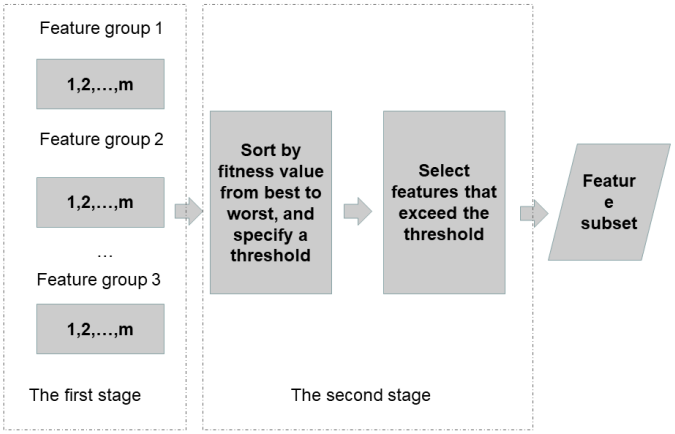

Figure 2: Feature selection framework based on ISFLA

The first stage: local optimization within the feature group.

The local search evolution in the feature subgroup relies on the worst individuals in the subgroup to randomly move toward the optimal individuals in the subgroup, so that the differences of all individuals in the group are gradually reduced, resulting in a gathering effect. Dynamically adjust the search step size and individual frog position using ISFLA algorithm.

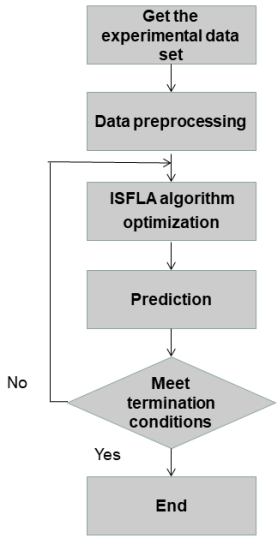

Figure 3: Local optimization process

The Second stage: global information interaction.

Sorting according to the fitness value, the ranking of the high fitness value (high correlation of features) is first. At the same time, a threshold is needed to select the final subset of features.

The threshold setting formula calculates the mean of all frog individual positions.

$$t = \frac{1}{n}\sum_{i=1}^{n} x_{(i)} \tag{6}$$

4 IoV defect prediction feature selection framework experiment based on ISFLA

The experiment divides into two parts: the first part verifies the validity of the framework using a recognized software measurement data set, and the second part builds an IoV system platform to verify the effectiveness of the framework in predicting the defects of the IoV terminal software.

4.1 Experimental data

In this experiment, we select the file-level data of the Eclipse standard data set [Zimmermann, Premraj and Zeller (2007)] and part of the data set in the software measurement database of NASA [NASA (2018)]. The Eclipse dataset is available from Eclipse Bug Data. The Eclipse dataset needs to be pre-processed before it can be used for defect prediction in this paper. According to a certain conversion rule, the conversion rules between the number of defects and posts are as follows:

$$\text{Defect}= \begin{cases} 0, & \text{post}=0 \\ 1, & \text{post}=1 \end{cases}$$

The experimental data after conversion, as shown in Tab. 1.

Table 1: Eclipse experimental data

Data set	Number of samples	Defect number
File2.0	6729	975
File2.1	7888	854
File3.0	10593	1568

The NASA experimental data, as shown in Tab. 2.

Table 2: NASA experimental data

Data set	Number of samples	Defect number
File2.0	6729	975
File2.1	7888	854
File3.0	10593	1568

4.2 Evaluation criteria

4.2.1 Accuracy

$$accuracy = \frac{TN + TP}{TN + FP + FN + TP}$$

Accuracy: The proportion of modules with correct classification results in all test cases. Where TP is a case (also known as a positive case) of correct classification. FN is a case of a misclassification, but it mistakenly considered as a positive case. TN is a case (also called a negative case) that has been correctly classified. FP is a case that is misclassified, but it mistakenly considered as a negative case.

4.2.2 Precision

$$precision = \frac{TP}{TP + FN}$$

Precision: the proportion of the number of defective modules whose prediction results are consistent with the actual situation to the predicted defective modules.

4.2.3 Recall

$$recall = \frac{TP}{TP + FP}$$

Recall: The probability of a defective module is predicted in the actual situation.

4.2.4 F-measure

$$F-measure=\frac{2}{\frac{1}{precision}+\frac{1}{recall}}$$

F-measure: Harmonic average of precision and recall.

4.3 Experimental results and analysis

Deep Forest [Zhou and Feng (2017)] (Multi-Grained Cascade Forest, geForest) is a software defect prediction method commonly used at present. The deep forest algorithm is a new type of deep structure learning algorithm, which opens up a new one in addition to deep neural network algorithms. It can be seen from Tab. 3 that in the four evaluation indexes of accuracy, precision, recall and F-measure, the improved SFLA method proposed in this paper has different degrees of performance improvement compared to the geForest method, indicating that this method has better defect prediction ability for file-level data under Eclipse experimental data.

Table 3: Defect prediction results of geForest and ISFLA under eclipse experimental data

geForest	Accuracy	Precision	Recall	F-measure
File2.0	0.8845	0.6708	0.4031	0.5036
File2.1	0.8555	0.3208	0.2998	0.3099
File3.0	0.8505	0.4912	0.2832	0.3593
ISFLA	Accuracy	Precision	Recall	F-measure
File2.0	0.8857	0.6710	0.4502	0.5389
File2.1	0.8565	0.3214	0.3013	0.3110
File3.0	0.8513	0.4917	0.2914	0.3659

Cluster Analysis [Xia, Hu and Luo (2017)] and back propagation (BP) neural network are currently common prediction methods of software defect. It can be seen from Figs. 4, 5, 6, and 7 that for some data sets of NASA, the ISFLA method proposed in this paper has different degrees of improvement in prediction performance compared with traditional BP neural network and CA in accuracy, precision, recall and F-measure. It shows that ISFLA cannot only select a smaller subset of features, but also has the advantage of further removing irrelevant features. It shows that this method also has a good defect prediction ability for some data sets of NASA.

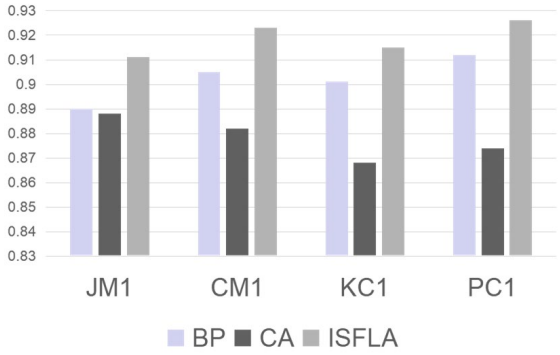

Figure 4: Comparison of accuracy

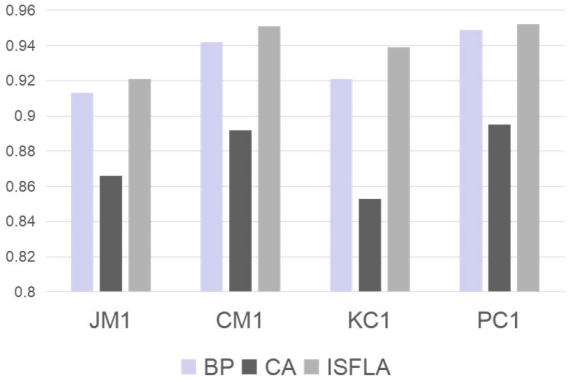

Figure 5: Comparison of precision

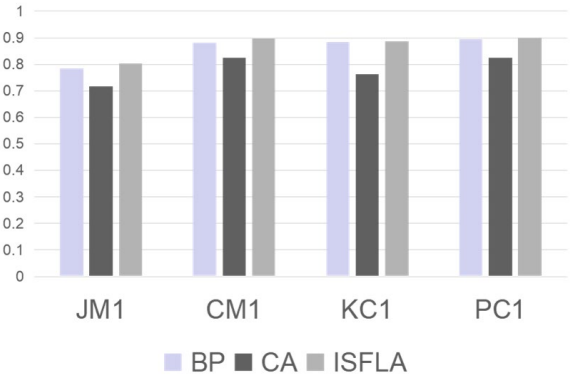

Figure 6: Comparison of recall

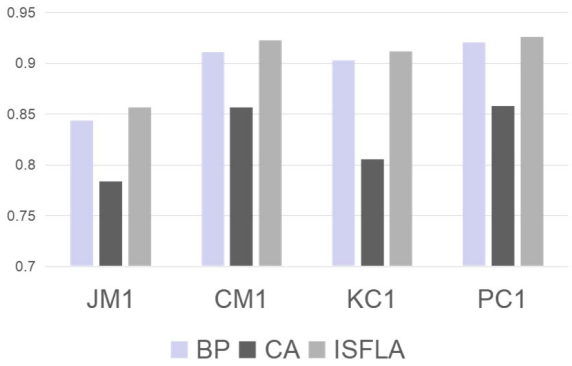

Figure 7: Comparison of F-measure

4.4 Test of IoV system platform

At present, the research on the performance and function of the Internet of Vehicles can be divided into software simulation analysis and building a system platform for testing [Zhang, Li and Wang (2018)]. In this paper, a system platform is for testing. The terminal software runs on the STM32F407 chip and is connected to the vehicle network through CAN Bus. The high-speed CAN Bus is mainly connected to the power system control unit. The low-speed CAN Bus is mainly connected to the car body comfort system such as central control lock, car doors and windows. The terminal equipment is connected to the CAN Bus of the vehicle to realize the perception and control of the vehicle. The communication software connects the on-board T-Box to the Internet through the 4G long term evolution (LTE) module, and reports the real-time status data of the vehicle to the connected vehicle platform. The connected vehicle platform can also actively issue instructions to the T-Box to control the vehicle through the communication software.

The essence of the Internet of Vehicles is the cross fusion of the automobile's internal network and the mobile Internet [Li (2017)]. The complexity of the Internet of Vehicles system itself determines that it must be tested and verified from multiple levels, dimensions, and perspectives. The above characteristics determine that testing in the Internet of Vehicle system is a complex and professional task [Chai, Cai, Wang et al. (2017)]. From the perspective of the test content, it mainly tests the information interaction between vehicle and cloud platform, that is, remote control and data collection. Remote control is mainly to issue control instructions to the vehicle-networking platform, including switch control of doors, windows, air conditioners, lights, trunks, etc. Data collection mainly includes the acquisition of vehicle body status (air condition, central control lock, windows, etc.) and driving data (speed, mileage, gear, etc.).

Test plan design:

Take the vehicle-networking test of a certain vehicle model as an example. The vehicle networking system of this vehicle model is composed of T-Box, 4G communication module, body control module (BCM) system, operator network, mobile APP, and service platform, which can realize the remote control and information exchange functions of mobile app to the vehicle, and its vehicle networking system architecture, as shown in Fig.

8. The test scheme design of the Internet of vehicle system, as shown in Tab. 4.

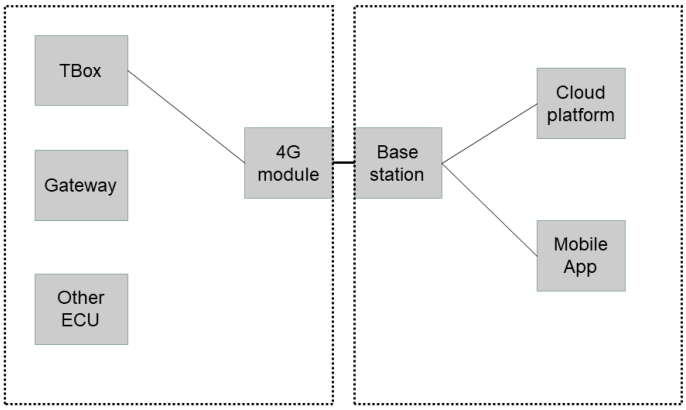

Figure 8: Architecture of internet of vehicles system

Table 4: Test scheme design

Category	Test content
Information exchange	Interactive test of vehicle body status and driving data information
Remote control	Control function test

The samples number derives from the historical test data of the first 10 iterations, as shown in Tab. 5.

Table 5: Experimental data of IoV

Data set	Number of samples	Defect number
Body status	1058	525
Driving data	1324	603
Remote control	2103	1121

Figs. 9, 10, 11 and 12 show the effects of defect prediction experiments using Accuracy, Precision, Recall, and F-measure on the IoV system platform. From the above experimental results, we found that the overall performance of ISFLA-based defect prediction feature selection framework in this paper has exceeded expectations on the three data sets. The experimental results of the four indicators on the remote control data set are all lower than the actual situation. After analysis of the experimental results, we found that the remote control refactoring, and the control mode is changed from comfort

network to comfort network and OBD collaborative control, with a large amount of changes and decreased stability.

Figure 9: Comparison of accuracy on the 11th iteration

Figure 10: Comparison of precision on the 11th iteration

Figure 11: Comparison of recall on the 11th iteration

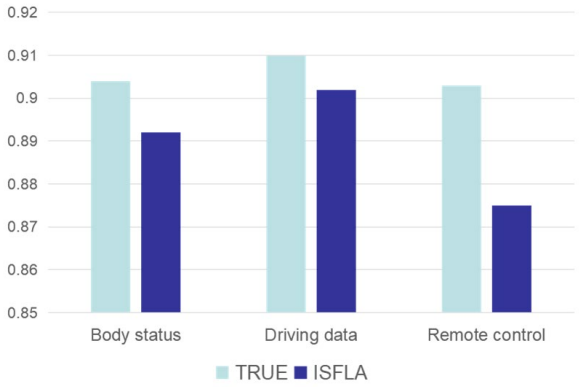

Figure 12: Comparison of F-measure on the 11th iteration

Fig. 13 shows the comparison between the prediction results and the actual results after applying the framework proposed in this paper on multiple iterations. The overall performance of the ISFLA defect prediction framework of feature selection in this paper has reached expectations in multiple iterations. The defect prediction feature selection framework proposed in this paper eliminates the erroneous data, redundant data and irrelevant measurement attributes in the defect prediction in the early stage of the framework, thereby reducing the dimensionality of the input data. In the later stage of the framework, we propose corresponding solutions for the two difficulties of feature selection, which improves the ability of defect prediction. Aiming at the dimensional disasters in software defect prediction research, and software complexity metric attributes and existing feature selection algorithms that are not related to defect prediction cannot satisfy the global optimal problem, this paper proposes a feature selection framework of defect prediction based on meta-heuristic search algorithm. Starting from the combinatorial optimization problem, the framework reduces the size of the data set and improves the accuracy of classification.

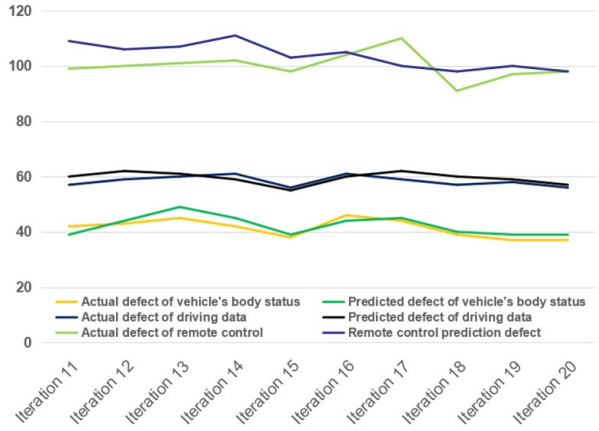

Figure 13: Comparison of prediction effects on each iteration

5 Conclusions and future work

To address the difficulty of feature selection and large search space in the process of feature selection of software defects, this paper proposes a feature selection framework of software defect prediction based on ISFLA algorithm. This research validates it in file-level data of Eclipse and some data sets of NASA, and then apply it to the IoV system platform for multiple rounds of iterative verification. Experimental results show that the performance of the framework achieves the expected results for defect prediction. There are still many places to be further explored and studied in this research work, including: a) there is a class imbalance problem in the data set itself, and the next work needs to further modify the prediction framework for this problem, so as to better alleviate this problem for defect prediction performance. b) In the next stage of work, try other algorithms to help the prediction framework to improve its prediction performance. c) This time, it mainly focuses on the defect prediction of the IoV communication software, and the next step is to predict the related defects of the IoV terminal software.

Funding Statement: This work was supported by Liaoning Natural Fund Guidance Plan Project (No. 20180550021), Dalian Science and Technology Star Project (No. 2017RQ021) and 2019 Qingdao Binhai University-level Science and Technology Plan Research Project (No. 2019KY09).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Chai, L. G.; Cai, B. G.; Wang, H. S.; Shangguan, W.; Wang, J. (2017): Simulation test method for impact of key indicators of the Internet of vehicles on vehicle safety. *Automotive Engineering*, vol. 39 no. 11, pp. 1316-1324.

Chang, X. G.; Zhao, H. S. (2016): Hybrid leapfrog algorithm based on trigonometric function search factor. *Computer Engineering and Science*, vol. 38, no. 11, pp. 2363-2364.

Duan, X. T.; Wei, J. Y.; Tian, D. X.; Zhou, J. S.; Xia, H. Y. et al. (2019): Adaptive handover decision inspired by biological mechanism in vehicle ad-hoc networks. *Computers, Materials & Continua*, vol. 61, no. 3, pp. 1117-1128.

Huang, M. J. (2013): Research on internet of vehicles and its application in intelligent transportation. *Applied Mechanics and Materials*, vol. 55, no. 2, pp. 321-324.

Ju, C. E.; Zhao, X. X.; Wen, Y. L.; Xing, R. (2018): Research on the communication mode of the Internet of vehicles based on the urban public transport cloud. *Information Technology*.

Kabir, M. M.; Shahjahan, M.; Murase, K. (2012): A new hybrid ant colony optimization algorithm for feature selection. *Expert Systems with Application*, vol. 39, no. 3, pp. 3747-3763.

Li, J. L.; Liu, Z. H.; Yang, F. C. (2014): Telematics architecture and key technologies. *Journal of Beijing University of Posts and Telecommunications*, vol. 37, no. 6, pp. 95-100.

Li, Z. T. (2017): Research and analysis of internet of vehicles test. *Automotive Appliances.*

Lin, H. L.; Huang, X. P.; Pasha, F. (2014): Review of research on telematics. *Mechanical and Electrical Engineering*, vol. 31, no. 9, pp. 1235-1238.

Liu, L. Q.; Wang, L. G.; Huo, J. Y.; Han, J. Y.; Liu, C. Z. (2014): Hybrid leapfrog algorithm based on fuzzy threshold compensation. *Computer Engineering*, vol. 40, no. 5, pp. 168-172.

Liu, Y.; Liu, L. F. (2013): Review of 802.11p-based internet of vehicles transmission protocol research. *Computer Engineering and Design*, vol. 34, no. 9, pp. 3007-3012.

Ma, J. (2016): Research on key technologies and applications of Internet of Vehicles. *Jiangsu Science and Technology Information*, vol. 1, no. 24, pp. 50-52.

NASA (2018): *The NASA Metrics Data Program.*

Wang, Q.; Wu, S. J.; Li, M. S. (2008): Software defect prediction technology. *Journal of Software*, vol. 19, no. 7, pp. 1565-1580.

Wang, R. M.; Deng, X. F.; Xu, Z. G.; Zhao, X. M. (2019): Survey on simulation testing and evaluation of internet of vehicles. *Application Research of Computers*, vol. 36, no. 7, pp. 1922-1925.

Wang, S. L.; Jiang, F.; Gu, Y. Y.; Zhuang, J. Y. (2018): Internet of vehicles test research based on Tbox test. *Automotive Appliances*.

Wang, X.; Yao, T. T.; Han, S. S.; Cao, D. P.; Wang, F. Y. (2018): Parallel vehicle networking: ACP-based intelligent vehicle network management and control. *Acta Automatica Sinica*, vol. 44, no. 8, pp. 1391-1404.

Wei, Y.; Wang, Q. Y. (2018): Analysis and development status of C-V2X cellular car networking standards. *Mobile Communications*.

Wu, T. Q. (2016): Analysis and prospect of the development of telematics. *Automotive Industry Research*, vol. 4, pp. 10-15.

Xia, Z. Q.; Hu, Z. Z.; Luo, J. P. (2017): UPTP vehicle trajectory prediction based on user preference under complexity environment. *Wireless Personal Communications*, vol. 97, no. 3, pp. 4651-4665.

Yu, W. J.; Li, X. B.; Yang, H.; Huang, B. (2018): A multi-objective metaheuristics study on solving constrained relay node deployment problem in WSNS. *Intelligent Automation and Soft Computing*, vol. 24, no. 2, pp. 367-376.

Zhang, J. B.; Li, Z.; Wang, C. F. (2018): LTE network performance test and analysis for Internet of vehicles. *Computer Engineering*.

Zhou, Z. H.; Feng, J. (2017): Deep Forest: Towards an alternative to deep neural network. *IJCAI-17*, pp. 3553-3559.

Zhu, H. D.; Ge, W. C. (2017): Research on heterogeneous network convergence mechanism in the Internet of vehicles. *Communication Technology*.

Zimmermann, T.; Premraj, R.; Zeller, A. (2007): Predicting defects for eclipse. *International Workshop on Predictor MODELS in Software Engineering*.