

Retraction

Retracted: Fuzzy Testing Method of CAN Bus of Charging Pile Based on Genetic Algorithm

Security and Communication Networks

Received 31 January 2023; Accepted 31 January 2023; Published 8 February 2023

Copyright © 2023 Security and Communication Networks. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Security and Communication Networks has retracted the article titled “Fuzzy Testing Method of CAN Bus of Charging Pile Based on Genetic Algorithm” [1] due to concerns that the peer review process has been compromised.

Following an investigation conducted by the Hindawi Research Integrity team [2], significant concerns were identified with the peer reviewers assigned to this article; the investigation has concluded that the peer review process was compromised. We therefore can no longer trust the peer review process, and the article is being retracted with the agreement of the Editorial Board.


The authors do not agree to the retraction.

References

- [1] L. Chang, X. Wang, Li Tian, M. Song, and Z. Zhang, “Fuzzy Testing Method of CAN Bus of Charging Pile Based on Genetic Algorithm,” *Security and Communication Networks*, vol. 2022, Article ID 2745175, 11 pages, 2022.
- [2] L. Ferguson, “Advancing Research Integrity Collaboratively and with Vigour,” 2022, <https://www.hindawi.com/post/advancing-research-integrity-collaboratively-and-vigour/>.

Research Article

Fuzzy Testing Method of CAN Bus of Charging Pile Based on Genetic Algorithm

Chang Liu,¹ Xueqiong Wang,¹ Li Tian,¹ Song Mei ,² and Zhendong Zhang²

¹State Grid Hubei Electric Power Research Institute, Wuhan, China

²School of Cyberspace Security, Huazhong University of Science and Technology, Wuhan, China

Correspondence should be addressed to Song Mei; meisong@mail.hust.edu.cn

Received 25 February 2022; Revised 14 March 2022; Accepted 22 March 2022; Published 13 April 2022

Academic Editor: Muhammad Arif

Copyright © 2022 Chang Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the guidance of the new energy policy and the continuous expansion of the new energy market, electric vehicles are the development direction of the automotive industry in the future, and the electric vehicle charging infrastructure is an important guarantee for the use of electric vehicles, which plays a positive role in promoting the popularity of electric vehicles. CAN bus protocol is the communication protocol between charging pile and electric vehicle, and its security concerns the safety of electric vehicles. In this article, a CAN bus fuzzy testing method based on genetic algorithm is proposed to solve the security problem of charging pile CAN bus. In this method, genetic algorithm is added in the fuzzy testing process of CAN bus protocol, that is, genetic algorithm is introduced in the generation of fuzzy data to search the test case that best conforms to CAN bus protocol, so as to greatly improve the detection efficiency of CAN bus protocol.

1. Introduction

With the increase expansion of the new energy market, the sales volume of electric vehicle market shows an explosive growth. Although the combination of charging pile and electric vehicle can bring convenience to everyone's travel, it also faces various information security problems. CAN bus is mostly used in communication network of background master station system of electric vehicle charging pile controller [1]. The data information transmitted in the charging pile network may be intercepted, stolen, deciphering, passive attack, illegal impersonation, malicious tampering, and other malicious threats, and the internal network of charging pile basically does not take any security measures. Once intruded, the attacker can control the voltage of the charging pile and even modify the charging amount and other data at will, thus causing great losses to the charging pile and public security. However, at present, there is a lack of extensive research on the security of communication protocol between electric vehicles and charging piles at home and abroad. In order to cope with various possible attacks under the new situation, this article

proposes a fuzzy testing method based on genetic algorithm for the security of CAN bus protocol of charging piles. In this method, genetic algorithm is added to generate test data, optimizes it, and finally generates fuzzy data that is more in line with the protocol. Then, it sends the optimized test case to the CAN bus between the charging pile and the electric vehicle and find the potential safety problems in the charging pile CAN bus according to the response of the charging pile.

2. Present Status and Research Both in China and Abroad

CAN (Controller Area Network) is called controller area network. Its emergence is mainly to solve the data exchange between many controllers and measuring equipment in the vehicle. It is a serial communication network that effectively supports distributed control system or real-time control [2]. The CAN bus follows the ISO/OSI standard model, and its overall structure is divided into physical layer and data link layer [3]. However, the design of CAN bus protocol does not consider the problem of communication security. Its

communication content is not encrypted, so it is easy to be attacked by the network.

There have been many researches on the communication security of vehicle CAN bus. For example, a group of researchers hacked into a jeep by taking advantage of the loopholes in the CAN bus protocol and were able to remotely control the throttle and braking system. Zhang et al. [4] proposed an anomaly detection mechanism for vehicle-mounted CAN bus based on relative entropy, which can detect data flow anomalies caused by attacks on vehicle-mounted CAN bus network. Wu et al. [5] proposed a CAN bus message anomaly detection method based on random forest model, which can effectively detect the abnormal data on the bus. However, there is little research on the security of CAN communication protocol between electric vehicle and charging pile. Xu et al. [6] proposed a CAN bus attack method based on the maximum and minimum ant colony algorithm. Xiang et al. [7] put forward an information security test method of intelligent networked vehicle CAN bus based on fuzzy test, replay attack test, and DoS attack test, which can strengthen the information security detection of vehicles. However, this method is aimed at the CAN communication inside the vehicle, and it does not touch the safety of CAN communication between vehicle and pile. Zhou et al. [8] proposed an electric vehicle charging communication architecture based on heartbeat message monitoring mechanism, which can monitor the status of communication data or communication link in real time. However, it is relatively passive in CAN safety detection. By attacking the charging pile, the charging pile can have problems, such as shutdown and denial of service. Although some defects and loopholes can be found using fuzzy test method in CAN protocol detection, there are deficiencies in evaluating the completion of fuzzy test and the acceptance rate of test cases, which leads to low detection efficiency. In view of this, this article proposes a CAN communication fuzzy test method based on genetic algorithm. The genetic algorithm is used to optimize the generated test data, and then, it produces fuzzy data that is more in line with the protocol of CAN bus. While improving the efficiency of fuzzy test, it can find the abnormal situations in CAN bus more accurately.

3. Relevant Technical Analysis

3.1. CAN Protocol Data Frame Analysis. CAN bus communication frame is divided into data frame, remote frame, error frame, overload frame, and frame interval. Data frame is a frame carrying data from the transmitter to the receiver [3], which is structurally composed of 7 segments. According to the length of ID code of arbitration segment, it is divided into standard frame (can2.0a) and extended frame (CAN2.0B). The protocol of the standard frame is an 11 bit identifier. While the extended frame is compatible with the 11 bit ID identifier, it is extended upward to the 29 bit ID identifier [9]. CAN2.0B protocol is used for communication between electric vehicle battery management system (BMS) and charging pile. Figure 1 presents the CAN bus data frame structure.

- (1) Frame start and end are used to define a data frame, and any data frame contains these two segments.
- (2) The ID of the arbitration segment specifies the priority of the data frame. The smaller the ID value, the higher the priority.
- (3) The control end of standard frame is composed of extended frame flag bit IDE, reserved bit R0 and data length code DLC, and the control section of extended frame is composed of IDE, R1, R0, and DLC.
- (4) The length of the data segment is 0–8 bytes, which can ensure real-time performance and strong anti-interference ability.
- (5) The CRC segment consists of 15-bit CRC value and 1-bit CRC delimiter, which is used to verify the data.
- (6) The ACK segment will send an explicit level when all frame segments in front of the frame received by the receiving node are correct.

3.2. CAN Protocol Communication Process. CAN bus is a serial communication protocol, Nodes in CAN will broadcast their message data to all nodes in the whole CAN network. The transceiver in CAN is responsible for sending the received data to the controller in the same serial network. The CAN controller verifies and judges the data. If it is not the required data message, it will be discarded. Figure 2 presents the process of CAN processing data at the data link layer.

In the process of sending and receiving messages, the sending and receiving error counter will automatically increase or decrease the count according to the error type and node status, as follows:

- (1) Initialize the error counter to 0 on reset
- (2) Whenever CAN detects an error or failure, the receive error counter (REC) will increase by 1
- (3) REC plus 8
- (4) REC minus 1 if one frame of message is successfully received
- (5) When CAN detects an error or failure when the transmitter sends data, the transmission error counter TEC increases by 8
- (6) If one frame of message is successfully sent, TEC minus 1
- (7) If TEC is less than 127, TEC plus 8
- (8) If TEC is greater than 255, disconnect from CAN bus

3.3. Attack Analysis of CAN Bus. Attacks against CAN bus can be classified into the following three categories.

3.3.1. Snooping Data. Due to the design flaws of CAN bus broadcast mechanism and no authentication mechanism, attackers can obtain information of other nodes in the CAN bus network at will. In addition, CAN bus cannot guarantee the confidentiality of data, and attackers can easily obtain the

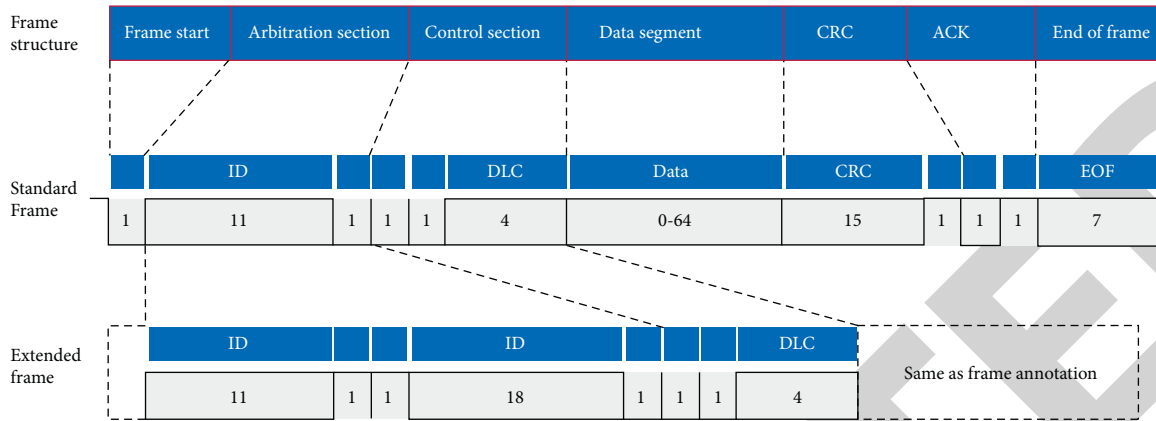


FIGURE 1: CAN bus data frame structure.

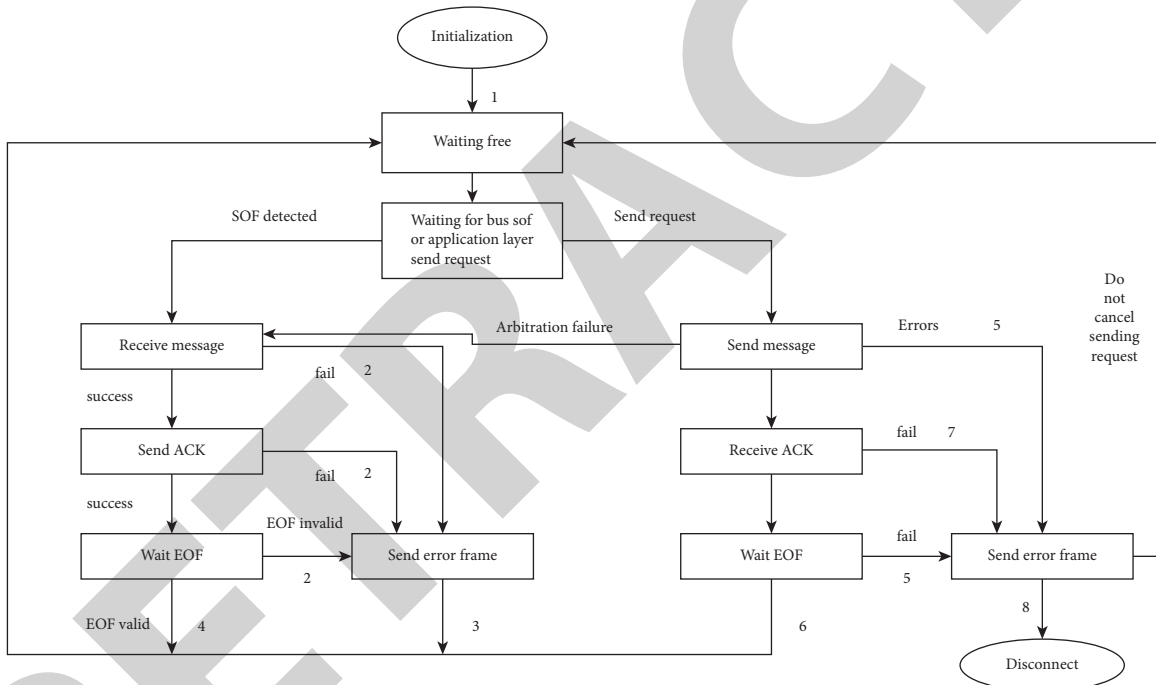


FIGURE 2: The process of CAN processing data at the data link layer.

communication information between each node in the network and perform the corresponding reverse analysis.

3.3.2. *Inject Malicious Data.* An attacker can inject a malicious message into the CAN bus network using an intruded or unauthorized node, and the corresponding receiving node will execute the malicious message. If malicious information is injected into the CAN communication between vehicle piles, it may cause serious consequences such as shutdown of charging piles.

3.3.3. *Flooding Data Attack.* Attackers can send huge amounts of messages to the CAN bus in a certain period, reducing the effective utilization of the bus. If the sending frequency is low, the receiving node will completely execute

according to the malicious message. If the transmission frequency is high, the low priority node will keep waiting for the bus to be idle, resulting in failure to communicate.

4. Fuzzy Testing Mechanism of CAN Bus Based on Genetic Algorithm

4.1. *Fuzzy Testing.* Fuzzy testing is a software testing technology. Its main principle is to generate abnormal test data through automatic or semiautomatic way and detect system vulnerabilities by repeatedly inputting the generated test data to the test target. It is one of the more practical vulnerability mining technologies at present [10, 11]. Compared with white box test and black box test, the test object of fuzzy test is binary object file, which can change the test data arbitrarily, and has good scalability and diversity. Fuzzy

testing is an automatic software vulnerability detection technology, which no longer requires manual reverse analysis, and can greatly reduce manpower and save time [12]. However, the traditional fuzzy test has the defects of low-generation efficiency and high failure rate in the test case placement. In order to solve this disadvantage, in the process of fuzzy data generation, genetic algorithm is introduced to search the test case that most conforms to the CAN protocol of charging pile, and a fuzzy test method based on genetic algorithm is designed. The six stages of fuzzy testing are shown in Figure 3.

- (1) Determine the test content, including test target type, historical vulnerability information, etc.
- (2) Determine the type of test data, such as data coding format, etc.
- (3) According to the input content, design the use case generation method and construct the use case set required by the test.
- (4) According to the data transmission protocol of CAN bus protocol, send test cases to the charging pile for fuzzy test. Among them, designing an appropriate fuzzer is important for fuzzy testing.
- (5) Set up a monitor to monitor the test run results, and integrate and analyze the test case information and abnormal conditions.
- (6) According to the final test results, determine whether there are potential or exploitable vulnerabilities and analyze them.

4.2. Genetic Algorithm

4.2.1. Basic Idea of Algorithm. In the process of fuzzy testing, genetic algorithm is added, and the idea of genetic algorithm is used to obtain the optimal test case. Through a specific coding method, the messages to be tested are integrated into a set of “genes” to form the initial test case group. Then, according to the idea of “natural selection and survival of the fittest” in the theory of evolution, through the inheritance of generations, the data test cases that most accord with the CAN protocol are generated. In the process of evolution, the fitness function is designed to control the generation of the next generation of use case population, and the genetic operator is used to cross and mutate the data, so as to generate a new generation of test case population. The optimal test case set in the last generation group is decoded to generate the most appropriate test case set. These test cases are mutated according to the preset rules to generate the final fuzzy data. In essence, it is an efficient, parallel, and global search method, which can automatically acquire and accumulate knowledge about the search space in the search process, and adaptively control the search process to obtain the best solution [13, 14]. Compared with other algorithms, genetic algorithm has the following advantages:

- (1) The probabilistic optimization method can automatically adjust the search direction and has randomness.

- (2) Directly operate the structure object, without the limitation of function continuity and derivation.
- (3) It has the ability of global optimization and implicit parallelism and can compare multiple individuals at the same time.
- (4) It is scalable and easy to combine with other algorithms.

4.2.2. The General Implementation of the Algorithm

- (1) Coding is to transform the parameters of the actual problem space into the form that genetic algorithm can deal with, and the coding method will have a direct impact on the subsequent steps such as crossover and mutation. Common coding methods include binary coding and gray coding. However, binary coding is not suitable for multidimensional and high-precision continuous function optimization problems, and there may be a large Hamming distance, so gray coding is selected [15].
- (2) In genetic algorithm, fitness function is used to judge the quality of individuals, and fitness function also plays a role in determining the direction of population evolution.

Firstly, a test case queue is created to store the optimal test cases calculated by genetic algorithm each time. The data of the initial queue is the CAN bus message data intercepted during the normal communication between electric vehicle BMS and charging pile. There are only two final classifications of CAN messages, namely, normal (expressed by $i=0.3$) and abnormal (expressed by $i=0.7$).

Then, after generating individual t_x , all individuals in the test case group with the same ID as individual t_x form a set S . Calculate the distance between t_x and each byte of individual S_i in set S , obtain the distance between t_x and S_i through the distance sum, and finally plus 1 to prevent the result from being 0. Then take the reciprocal and multiply by the status i of this message. The final result is $D(t_x, S_i)$, as shown in the following formula.

$$D_{t_x, S_i} = \frac{i}{\left(\sum_{i=0}^{l_x-1} (a_i^x - a_i^y)^2\right) + 1}$$

$$t_x = \{a_0^x, a_1^x, \dots, a_{l_x-1}^x\}, \quad (1)$$

$$S_i = \{a_0^y, a_1^y, \dots, a_{l_i-1}^y\}.$$

Finally, the distance between t_x and individuals in set S is added in turn and divided by the number n of set S to obtain the average distance $\overline{D_{t_x, S}}$ between t_x and the initial population. And take it as the fitness value of the individual, as shown in the following formula.

$$\text{fitness}(t_x) = \overline{D_{t_x, S}} = \frac{\sum_{j=0}^{n-1} D_{t_x, S_j}}{n}. \quad (2)$$

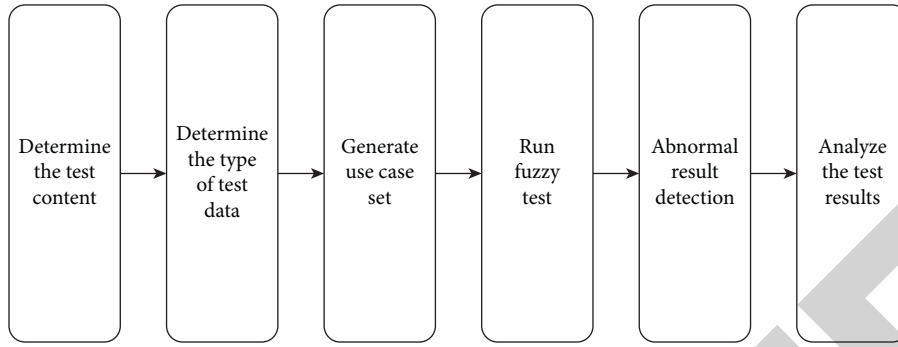


FIGURE 3: Six basic stages of fuzzy testing.

- (3) Select some chromosomes to produce the next generation. This article adopts the “roulette wheel selection” algorithm.

Firstly, the probability $P(m_i)$ of all individuals in the population being inherited is calculated according to the probability formula, as shown in the following formula.

$$P(m_i) = \frac{\text{fitness}(m_i)}{\sum_{j=1}^N m_j}, \quad (3)$$

probability Q_i of all individuals in the population, as shown in the following formula:

$$Q_i = \sum_{j=1}^i P(m_j). \quad (4)$$

Then, the random function $\text{rand}()$ is used to generate the random value r before $[0,1]$. If the random value is less than Q_1 , individual 1 is selected to enter the offspring population. Otherwise, select the individual k satisfying $Q_{k-1} < r < Q_k$ to enter the offspring population for the next operation. Finally, the individual obtained by repeating this step will form a new population.

- (4) Crossover refers to the recombination of some genes of two parent individuals in a certain way to generate a new individual.

This experiment adopts the way of single point crossover. Firstly, the individuals in the population are sorted according to the size of fitness value, and the individuals with large fitness value are preferentially selected for crossover. At the same time, in order to prevent the loss of the optimal solution, the individual with the largest fitness value does not participate in the cross operation of this round. The specific operation is to obtain a bit value using a random function on the gene sequence of two parent individuals (encoded and stored in binary mode). The two parent gene sequences are exchanged after the site to produce a new child. Figure 4 presents the cross method.

- (5) Mutation is to reverse one or several bits of the binary string corresponding to each individual, that is, from 1 to 0 and from 0 to 1. However, because variation is highly destructive [16], it is necessary to

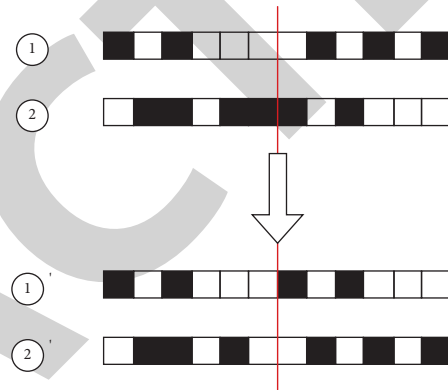


FIGURE 4: Cross method.

set a variation probability P , as shown in the following formula.

$$P = k \times \frac{n}{l} \quad k \in [0, 1]. \quad (5)$$

In this experiment, n is the number of individuals, l is each individual, and l is the length of the binary data string of the individual.

Similarly, in order to prevent losing the optimal solution, a threshold value should be set for the fitness value. If the fitness value of a chromosome is greater than the threshold value, the pos bit on the chromosome data string will be reversed. The formula of pos is shown in the following formula.

$$\text{pos} = \begin{cases} \text{random}(n) \times L + \text{random}\left(\frac{L}{i}\right) & f \geq f_{\text{avg}} \\ \text{random}(n) \times L + \text{random}(L) & f < f_{\text{avg}} \end{cases}, \quad (6)$$

where n is the number of individuals, L is the individual's own coding length, $i \in [1, L]$, and f is the fitness value of the current individual, using f_{avg} is the threshold value of fitness value in the above, which represents the average value of all individual fitness values in the population of this generation.

- (6) After several rounds of “genetic” screening, the optimal solution is sent to the charging pile as a test

case through the CAN bus fuzzy test system, and the feedback results of the charging pile are analysed.

4.3. CAN Bus Fuzzy Test System Based on Genetic Algorithm.

The fuzzy test system mainly has three new characteristics: CAN bus communication management, CAN data variation optimization, and detection result management. It can achieve CAN communication data monitoring, sending and receiving, multiprotocols synchronous detection, dynamic generation of test cases, convenient content query, and so on. At the same time, compared with the traditional fuzzy testing, this scheme introduces the improved genetic algorithm, and through the user-defined mutation scheme, it is used to generate test cases with higher coverage and verify the test samples causing abnormalities. This method significantly improves the efficiency and accuracy of CAN bus anomaly detection.

4.3.1. CAN Fuzzy Detection System Architecture. The system in this article takes Raspberry Pi 4B and its extended development board 2-CH CAN HAT as the main control chip to build a CAN bus receiving and transmitting platform, which can collect and process message data, and it is connected in series between the charging pile and BMS simulator. Figure 5 presents the CAN bus safety detection system architecture.

After the hardware access is completed, the baud rate of CAN bus protocol is uniformly configured and connected to the network. After the platform can normally receive the CAN bus signal, analyze, and forward the received message signal by writing Python script. Because the CAN protocol may be affected by the outside world in the transmission process, it is necessary to simulate the CAN protocol to send and receive messages with the help of analyst tool (the system adopts Chuangxin technology analyst - II analyzer), forward messages, and optimize the script written by yourself.

Although limited by the actual environment, it is not possible to select multiple models of electric vehicles for testing; the parameters of BMS simulator can be adjusted to simulate multiple brands and models of electric vehicles. In addition, the selected model of Raspberry Pi is 4B, which is common in the market. Except for their own debugging tools, they are all open-source software. Therefore, this system has high feasibility.

4.3.2. CAN Fuzzy Test System Function Module. The system is mainly written in C language. The core part is to use fuzzy test based on genetic algorithm to detect the security of charging pile CAN protocol. According to the classification of functional modules, it is divided into three parts. They are protocol management module, sample generation module, and result monitoring module. Figure 6 presents the module structure of fuzzy test system.

(1) Protocol Configuration Module. This module is mainly used to manage CAN protocol, including protocol name and

description. After the tested protocol is added, you can customize the data frame under the relevant protocol. The protocol configured in this module can be selected in the data variation optimization part, and it will send and receive message data under normal communication status. This part can also add, delete, edit, and view test tasks and test objectives. For example, for charging piles of different network segments, charging piles of different models and brands can carry out test tasks according to their own requirements.

(2) Sample Generation Module. As the core module of the system function, the sample generation module is used to carry out customized variation strategies after capturing the communication message data. The customized strategies affect the effectiveness of the test samples. This part mainly takes the normal communication data as the original sample, calls the genetic algorithm according to the user-defined mutation optimization strategy, optimizes the original data according to the idea of "survival of the fittest," and finally generates test cases with wider coverage and higher accuracy. This innovative method significantly improves the hit rate and execution efficiency of detection. The most important part is to configure the mutation strategy, which is directly related to the content of the sample. This section proposes a combination of predefined and customized mutation strategies, which can configure mutation strategies in a variety of ways and improve the accuracy and breadth of mutation samples, including the functions of adding, deleting, editing, and retrieving mutation strategies. Predefined is to generate samples by the mutation strategy preset by the system itself. Customization can customize key parameters, such as different fitness functions, selection factors, and variation factors to generate optimal test cases.

(3) Execute the Monitoring Module. The module is mainly used for test management, test execution, test status monitoring, and test result storage. This is the only part of the system that directly interacts with the target charging pile and is also responsible for task execution and scheduling. The module mainly includes the configuration of test tasks and test methods, query of test progress, query of variation details, and the like. It can manage the configuration of test tasks, view the execution progress of test tasks, and view and analyze the variation details of completed tasks, mainly including the correct recording and printing of sensitive data packets. The recorded information includes target information, test time, and abnormal behavior, which can facilitate the analysis and abnormal tracking of detection process and results.

4.3.3. The Workflow of CAN Fuzzy Test System. Figure 7 presents the workflow of the system.

The user enters the system through the WEB management interface and first carries out the relevant contents of the preparation stage, including configuring the CAN protocol, defining the variation strategy, transmitting the generation strategy to the sample generation module, constructing the variation sample by traversing the linked

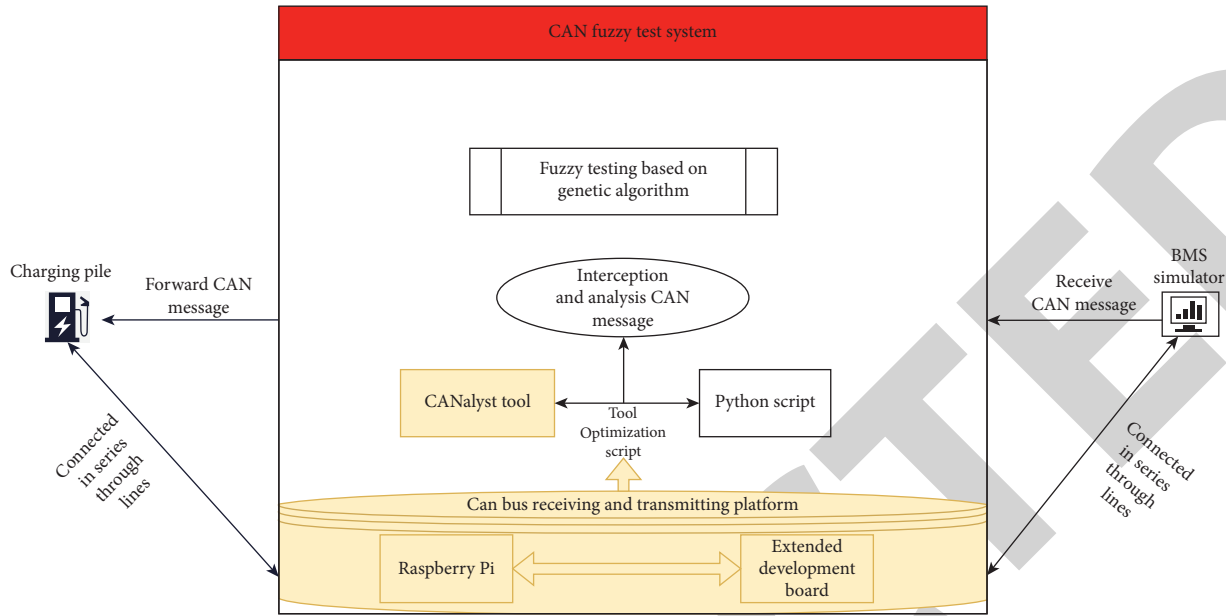


FIGURE 5: CAN bus safety detection system architecture.

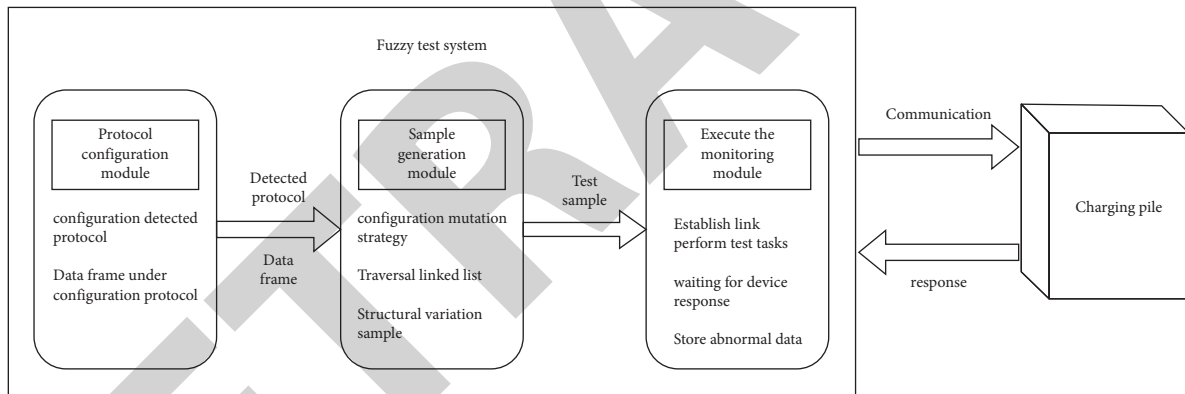


FIGURE 6: Module structure of fuzzy test system.

list, and transmitting the sample to the execution module for use after generation.

When the execution test starts, the task interacts with the background resident fuzzy detection main process. After receiving the start test notification, the process starts the test thread, which starts the specific execution of the test task, and notifies the foreground that the test task has been started.

In the execution process, first carry out link management. After establishing the link, start sending test samples to the tested equipment and wait for the response of the tested equipment. If the tested equipment responds normally, the sample test passes. If the inspected equipment fails to respond after timeout, it will be judged that the response state is abnormal, and the test sample data will be sent, marked, and recorded. If there are still test samples not sent, return to the step of sending data to continue sending data. If all the data have been sent, complete an execution stage, and

then enter the system to query the complete task execution related results.

5. Experiment and Analysis

5.1. Experimental Procedure. In the experiment, the whole charging communication mainly involves the battery management system and charging pile of electric vehicle. Among them, the battery management system of the electric vehicle is replaced by the SAISIN ST-880BMS simulator, and the charging pile is selected from the State Grid Pratt TCU-2100 DC charging pile. Both parties agreed on the power requirements as well as the amperage and voltage to be used during the charging process and monitored the charging process at each stage of the charging communication. If a fault occurs, the charging process can be terminated. The charger can then choose to restart the charging process. Using the GB/T-27930 protocol, communication is divided

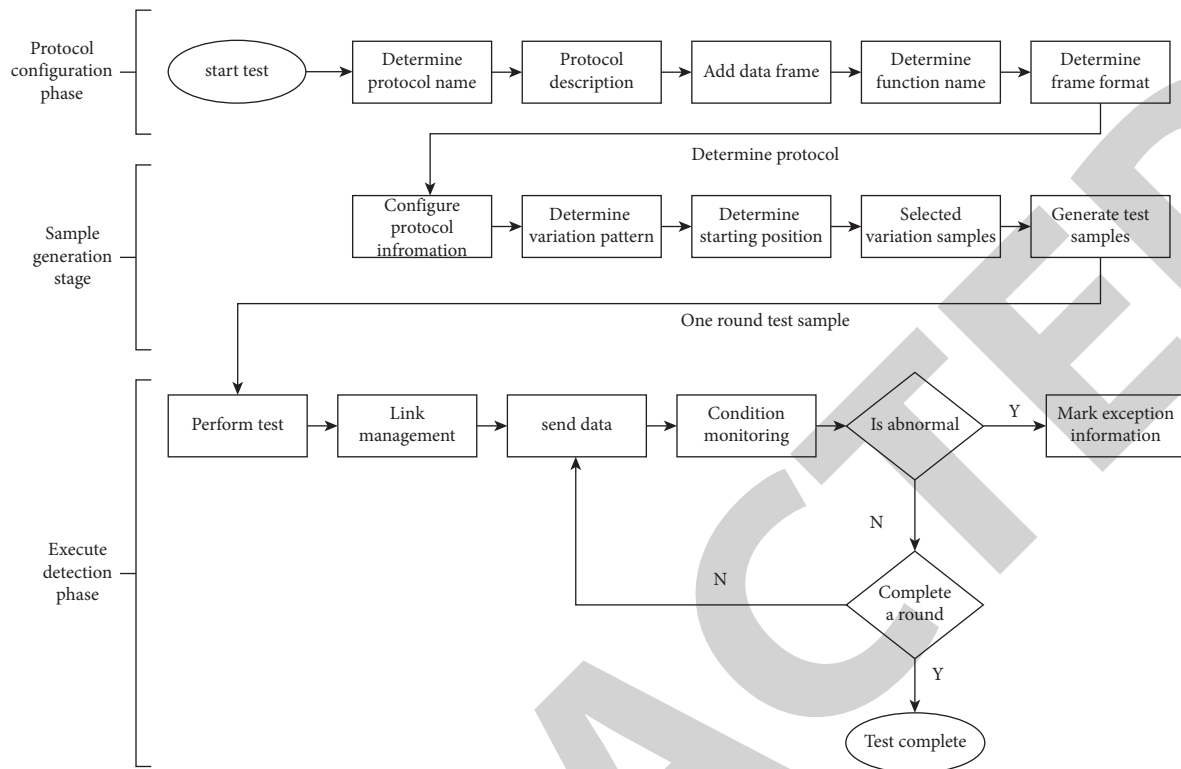


FIGURE 7: Workflow of the system.

into the following phases during the charging process, as shown in Figure 8.

5.1.1. Handshake Initiation. The first stage of communication begins once the system is connected in series between the charging pile and the BMS simulator. At this point, the charging process has not yet started, that is, the current has not yet flowed. The connection is checked and then the BMS simulator informs the charger of the maximum allowable charging voltage.

5.1.2. Handshake Identification. In the handshake identification phase, the charger connection check is completed and general information is exchanged, such as protocol version and vehicle information (battery type, vehicle identification number, etc.).

5.1.3. Parameter Configuration. In the parameter configuration stage, the parameters of the charging process are negotiated. The BMS simulator informs the charger of the allowed amperage and voltage. The charger informs the BMS simulator of the available amperage and charging voltage.

5.1.4. Charging. If the charger can meet the requirements of the BMS simulator, the charging process starts in the charging phase and the “battery” is charged. During the charging process, the BMS simulator periodically informs the charger of the current state of charge of the battery.

5.1.5. Suspended Charging. During the suspended charging phase, either party can terminate the charging process. Reasons for this could include the “battery” being fully charged or failing during charging.

5.1.6. End of Charge. At the end of the charge phase, the charger stops outputting power.

Because the charging pile is very sensitive to the ID of the message and some messages, once it finds that the received information does not meet the protocol standard, the charging pile will stop working immediately. Therefore, in order to improve the execution efficiency of the experiment, the detected packets need to be selected in the experiment, and the fuzz test is mainly carried out for the charging stage. The messages that need to be cross-mutated are shown in Table 1.

The message code in the table is the message code designed by the new national standard 27930–2015 of charging pile, and the Description column is its explanation.

5.2. Experimental Results. Use the self-written fuzzing tool to continuously test the charging process of the charging pile, and you can find errors in the charging pile from the intercepted packets. Table 2 is a part of the error message obtained in the test.

It can be seen from Table 2 that before, the charging pile and the BMS have been communicating normally. Starting from the 19374th message, it is the error message information sent by the charging pile to the BMS after detecting an error. After that, the charging pile starts to send 1801F456

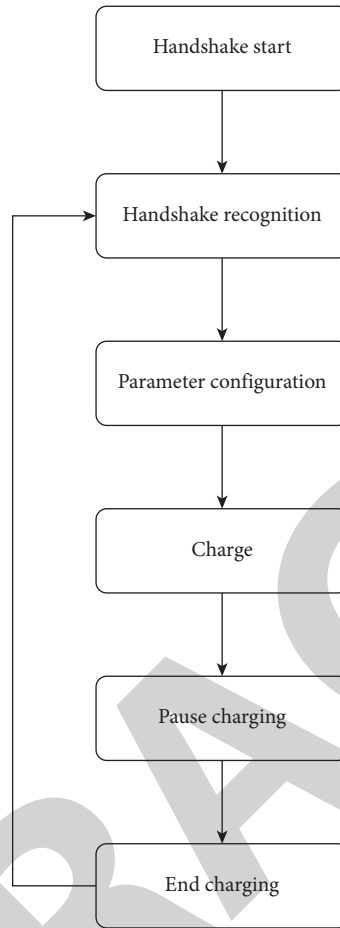


FIGURE 8: Communication process during charging.

TABLE 1: Cross message variation.

Phase	Message code	Description	PGN	PGN(Hex)	Data length(byte)	Priority
Charging phase	BCL	Charging needs	4096	001000H	5	6
Charging phase	BCS	Battery charge status	4352	001100H	9	6
Charging phase	CCS	Charger charging status	4608	001200H	6	6
Charging phase	BSM	BMS battery information	4864	001300H	7	6
Charging phase	BMV	Single cell voltage	5376	001500H	Indefinite	6
Charging phase	BMT	Power battery temperature	5632	001600H	Indefinite	6
Charging phase	BST	BMS aborts charging	6400	001900H	4	4
Charging phase	CST	Charger stops charging	6656	001A00H	4	4

TABLE 2: Partial message information.

No.	Message code	Frame type	Frame format	ID	Date length (bytes)	Phase
19368	BRO	Date frame	Extended frame	100956F4	1	Parameter configuration
19369	CRO	Date frame	Extended frame	101AF456	1	Parameter configuration
19370	BCL	Date frame	Extended frame	181056F4	5	Charging phase
19371	BCS	Date frame	Extended frame	1CEC56 F	9	Charging phase
19372	CCS	Date frame	Extended frame	1812F456	6	Charging phase
19373	CCS	Date frame	Extended frame	1812F456	6	Charging phase
19374	CEM	Date frame	Extended frame	081FF456	4	Error
19375	CEM	Date frame	Extended frame	081FF456	4	Error
19376	CRM	Date frame	Extended frame	1801F456	8	Handshake phase
...	CRM	Date frame	Extended frame	1801F456	8	Handshake phase
19380	CST	Date frame	Extended frame	101AF456	4	Charging phase

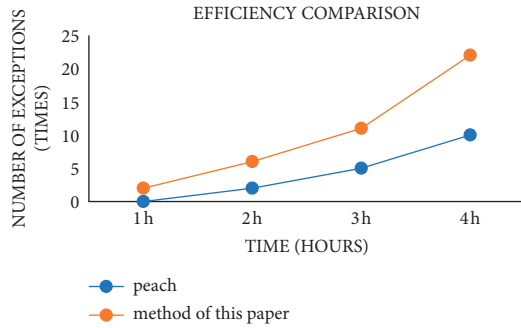


FIGURE 9: Efficiency comparison.

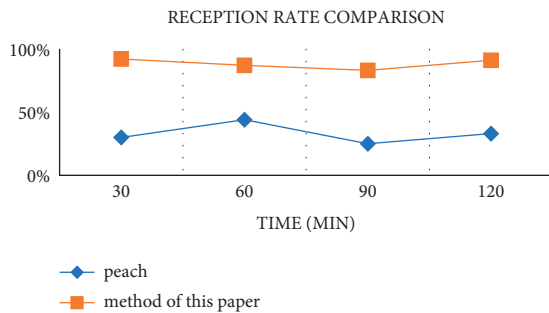


FIGURE 10: Reception rate comparison.

(hexadecimal) messages, that is, it returns to the handshake stage for identification. The header of the message is 00, which means it cannot be recognized (AA means it can be recognized). After the continuous unrecognized, the charging pile sends a termination message, and it was unable to respond to subsequent operations. This indicates that this experiment caused the charging pile to go down and also achieved the purpose of the fuzzy test.

5.3. Experiment Analysis. In order to verify the test efficiency and test data reception rate of the fuzzing method proposed in this article with the addition of genetic algorithm, this study uses a comprehensive open-source fuzzing tool peach to compare with the experimental method, aiming at the CAN communication protocol of the same charging pile. Safety checks were carried out. Figures 9 and 10 list the time efficiency comparison and test case absorption rate comparison of the two methods, respectively.

It can be seen from Figure 9 that within the same period, using the fuzzing test method in this article, more test cases that conform to the CAN protocol specification can be generated faster and more, which will lead to abnormal charging piles. Compared with the fuzzing method of the peach tool, the detection efficiency is greatly improved. It can be seen from the results in Figure 10 that in the first two hours of interception, the fuzzing method used in this article is much higher than peach in terms of test case acceptance rate.

6. Conclusion

In this experiment, a genetic algorithm was added to the fuzzing test of the charging pile CAN bus protocol, which

improved the accuracy and detection efficiency of the fuzzing test. Because the CAN bus lacks a basic security mechanism and has a certain vulnerability, it is found that the charging pile can be shut down after processing some messages such as crossover and mutation. At the same time, due to the poor self-recovery ability of the charging pile, some errors will cause the charging pile to stop working directly, requiring manual restart, causing great inconvenience to users and charging pile manufacturers. Based on this detection method, the vulnerability of the CAN protocol can be used to perform safety detection on the charging pile, and potential information security problems can be found, thereby improving the safety and reliability of the electric vehicle during the charging process.

Due to the wide range of brands and models of electric vehicles, this experiment could not be exhaustive. In the future, we will continue to improve and optimize the genetic algorithm to further improve the detection efficiency of the fuzzy test system.

Data Availability

The data sets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors acknowledge the National Natural Science Foundation of China (Grant: 62072200).

References

- [1] J. Liu and Y. Gao, "The application of CAN bus for communications between the electric vehicle's BMS manage system and charge picket," *Microcomputer information*, vol. 000, no. 009, pp. 205–207, 2012.
- [2] G. Wang and J. Qian, "CAN bus and the higher layer protocol based on CAN Protocol," *Computer Measurement and Control*, vol. 11, no. 5, pp. 391–394, 2003.
- [3] J. Jiang, Y. Lin, and J. Han, "Analysis and implementation of communication protocol of CAN," *Computer Engineering*, vol. 28, no. 2, pp. 219–220, 2002.
- [4] H. Zhang, rongshuai Jiang, J. Wang, Z. Lu, and Z. Liu, "Research on anomaly detection of in-vehicle CAN bus based on entropy," *Automotive Engineering*, vol. 43, no. 10, pp. 1543–1548, 2021.
- [5] L. Wu, G. Qin, and He Yu, "Anomaly detection method of in-vehicle CAN bus based on random forest," *Journal of Jilin University: Science Edition*, vol. 56, no. 3, pp. 663–668, 2018.
- [6] J. Xu, J. Wang, L. Chang, L. Zhou, and L. Feng, "Security detection of CAN bus protocol for electric vehicle charging pile," *Journal of Shandong University: Science Edition*, vol. 55, no. 5, pp. 95–104, 2020.
- [7] M. Xiang, W. Tianwen, Yu Deng, and S. Liu, *Can Bus Cyber Security Test Method of Intelligent Connected Vehicle*, Times automobile, no. 20, , pp. 197–198, 2021.

- [8] C. Zhou, Z. Wang, Z. Li, F. ruijue, S. Yang, and X. Zeng, "Research on CAN message of electric vehicle charging based on heartbeat mechanism," *Environmental Technology*, vol. 39, no. 3, pp. 181–185, 2021.
- [9] Y. Wang, Y. Li, X. Chen, L. Liu, and min Wu, "Replay attack and defense method of CAN protocol for electric vehicle charging," *Journal of Shanghai Electric Power University*, vol. 37, no. 4, pp. 395–401, 2021.
- [10] Tewodros Legesse Munea, I. L. Kim, and T. Shon, "Design and implementation of fuzzing framework based on iot applications," *Wireless Personal Communications*, vol. 93, no. 2, pp. 365–382, 2017.
- [11] U. Kargén, Shahmehri, and Nahid, "Turning programs against each other: high coverage fuzz-testing using binary-code mutation and dynamic slicing," *Joint Meeting on Foundations of Software Engineering*, pp. 782–792, 2015.
- [12] X. Liu, B. Cui, J. Fu, and J. Ma, "Hfuzz: towards automatic fuzzing testing of nb-iot core network protocols implementations," *Future Generation Computer Systems*, vol. 108, no. Jul, pp. 390–400, 2020.
- [13] Z. Bao, K. Chen, Z. he, and C. Han, "A sparse circular arrays method based on modified genetic algorithm," *Electronic technology application*, vol. 34, no. 10, pp. 110–112, 2008.
- [14] D. W. Coit, "Genetic algorithms and engineering design," *The Engineering Economist*, vol. 43, no. 4, pp. 379–381, 1998.
- [15] Li Xiang and R. Ma, "Research on Fuzzy testing technology of Modbus TCP protocol based on genetic algorithm," *Ship Electronic Engineering*, vol. 40, no. 10, pp. 149–153, 2020.
- [16] C. Chen and N. Wang, "Adaptive selection of crossover and mutation probability of genetic algorithm and its mechanism," *Control theory and application*, vol. 19, no. 1, pp. 41–43, 2002.