Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2021.017214
Article

Tech Science Press

Case Optimization Using Improved Genetic Algorithm for Industrial Fuzzing Test

Ming Wan[1], Shiyan Zhang[1], Yan Song[2], Jiangyuan Yao[3,*], Hao Luo[1] and Xingcan Cao[4]

[1]School of Information, Liaoning University, Shenyang 110036, China
[2]School of Physics, Liaoning University, Shenyang 110036, China
[3]School of Computer Science & Cyperspace Security, Hainan University, Haikou 570228, China
[4]Faculty of Arts, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada
[*]Corresponding Author: Jiangyuan Yao. Email: yaojy@hainanu.edu.cn
Received: 24 January 2021; Accepted: 27 February 2021

Abstract: Due to the lack of security consideration in the original design of industrial communication protocols, industrial fuzzing test which can successfully exploit various potential security vulnerabilities has become one new research hotspot. However, one critical issue is how to improve its testing efficiency. From this point of view, this paper proposes a novel fuzzing test case optimization approach based on improved genetic algorithm for industrial communication protocols. Moreover, a new individual selection strategy is designed as the selection operator in this genetic algorithm, which can be actively engaged in the fuzzing test case optimization process. In this individual selection strategy, the selection operation based on high and low fitness populations is introduced to enhance the individual selection diversity, which can increase the average fitness value of individuals and further improve the efficiency of test cases. In practice, we construct industrial communication data which conforms to Siemens S7 communication protocol to evaluate the proposed approach, and the experimental results show that, the individual fitness value of output population in the improved genetic algorithm is obviously higher than the one in traditional genetic algorithm under the same iteration, and this approach can enhance the efficiency and accuracy of test cases in Siemens S7 fuzzing vulnerability exploiting.

Keywords: Industrial fuzzing test; improved genetic algorithm; test case optimization; vulnerability

1 Introduction

With the gradual integration of OT (Operation Technology) and ICT (Information Communication Technologies) [1], today's industrial control systems are rapidly changing from the self-closed information island to the wide-open interconnection architecture, and this situation can greatly increase the risk of industrial cyber attacks [2]. As a result, various information security incidents have occurred in recent years, and industrial cyber security has gradually become the focus of attention all over the world [3–6]. As one significant component in industrial control systems, industrial communication protocols can

be regarded as one critical communication medium which can realize the function of remote control and automation [7]. However, the basic design and implementation of different industrial communication protocols may not only have certain flaws, but also miss enough security considerations. In other words, the vulnerabilities of industrial communication protocols have become one of the most attractive targets for organized cyber attacks [4,8,9]. In order to successfully exploit various potential security vulnerabilities in industrial communication protocols, both industry and academia have started to carry out some related researches on industrial security testing technologies [10–13].

Different from traditional IT communication protocols, industrial communication protocols have their specific characteristics [14,15]. Moreover, these specific characteristics not only reflect in the unique design of protocol specifications, but also are embedded in different protocol developments and implementations provided by various industrial device manufacturers. As a result, all of these may present a challenge for the vulnerability exploiting of industrial communication protocols [16]. Although many researchers have proposed some security scanning technologies for known vulnerabilities of industrial control devices [17], the prerequisite for their successful execution needs to rely on the comprehensiveness of disclosed vulnerability information, which has great restraints on their application. As one frequently-used black-box testing technology in traditional IT systems, fuzzing test which is also considered as a robustness testing approach can realize the automatic vulnerability mining by using injection flaws [12,18–20]. Furthermore, this technology can not only remark the existing known flaws with little or no trouble, but also have the potential to exploit some zero-day flaws. Due to the advantages of simple execution and high automation, fuzzing test may be further developed into practical applications in the vulnerability exploiting of industrial communication protocols [21–23]. More specifically, by dynamically injecting the distorted data into one target object, this technology can detect some possible attack entrances without considering the implementation details and complexity of target object. Additionally, it can be smoothly integrated with random testing, fault injection and grammar deviation, and focus on the deep data mining to explore the security defects of target object. In general, the test case generation, which can generate all kinds of distorted data to construct test cases, is a critical step in the whole fuzzing test framework [12,24]. However, for complex industrial control communication protocols, it may produce too many redundant test cases under the exhaustive or blind construction, and cause the unsatisfied execution efficiency of fuzzing test.

Aiming at optimizing the way to generate multiple and serviceable distorted data, this paper proposes a novel fuzzing test case optimization approach based on improved genetic algorithm for industrial communication protocols. Moreover, this approach designs a new individual selection strategy for the traditional genetic algorithm to carry out the selection operation. In this individual selection strategy, the selection operation based on high and low fitness populations is introduced to enhance the individual selection diversity, which can increase the average fitness value of individuals and further improve the efficiency of test cases. Additionally, by using the initial data which conforms to Siemens S7 communication protocol, this approach can generate more effective test cases through the corresponding mutation strategy. The experimental results and compared analysis show that, the individual fitness value of output population in the improved genetic algorithm is obviously higher than the one in traditional genetic algorithm under the same iteration, and this approach can enhance the efficiency and accuracy of test cases in Siemens S7 fuzzing vulnerability exploiting.

The main contributions of this paper are summed up as follows: firstly, we analyze the basic fuzzing test framework for industrial communication protocols, and state the need and purpose of test case optimization; Secondly, we propose a novel genetic algorithm to effectively optimize fuzzing test cases, and cover the detailed descriptions on the design and assumption of this algorithm; thirdly, based on Siemens S7 communication protocol, we not only give a compared evaluation on the optimization performance of

this approach, but also discuss different influences of critical parameters to further meet different industrial communication protocols and optimization requirements.

2 Basic Fuzzing Test Framework for Industrial Communication Protocols

In various industrial control activities, industrial communication protocols can be regarded as one important medium to realize real-time control and remote management between different industrial control devices, but they still retain some substantial security flaws. Differently, fuzzing test has an amazing ability to discover these flaws for industrial communication protocols, which can make the target object crash by constructing all kinds of irregular protocol data. To be precise, fuzzing test has become one of the most practical vulnerability exploiting approaches for industrial communication protocols in the field of industrial cyber security.

As shown in Fig. 1, the basic fuzzing test steps for industrial communication protocols are summarized as follows:

(1) Select one industrial communication protocol as the test target.

(2) Build the appropriate communication environment according to the requirements of selected protocol.

(3) Understand the data format and protocol specification of the selected industrial communication protocol, and parse the initial communication data.

(4) Based on the model-based or mutation-based generation approach, generate and optimize the practicable fuzzing test cases.

(5) Send the optimized test cases to the targeted object according to the transmission specification and working mode of the selected industrial communication protocol.

(6) Monitor the feedback results which can reflect the running status of target object. If no anomaly is found, a new test case will be sent; otherwise, the test case and the corresponding feedback results will be further analyzed to identify the potential vulnerabilities.

(7) Analyze the main cause according to the abnormal results, and record the potential vulnerabilities.

In the fuzzing test framework, the test case generation is one critical step which can determine the quality of vulnerability exploiting, because one fine test case generation approach can explore more security defects when the number of test cases is limited. In the process of test case generation, a large number of unexpected semi-structured industrial communication data are generated according to the specified data format and protocol specification. In general, the fuzzing case generation approaches can be divided into two categories: the model-based generation approach [25] and the mutation-based generation approach [26]. Furthermore, the model-based generation approach can build one rule model which conforms to the selected industrial communication protocol to generate original test cases. However, this test case generation approach requires a deep understanding of the selected industrial communication protocol. Differently, the mutation-based generation approach can construct new test cases by inserting diversified distortion bytes into normal data, which is generally realized by the mutation algorithm. However, this test case generation approach also has some shortcomings: the test efficiency of test cases generated by the mutation algorithm is relatively low, and the fuzzing test has to end up with unexpected results. Additionally, it seems difficult to guarantee the test coverage. In order to exploit more vulnerabilities as effectively as possible, a necessary and feasible way is to optimize the generated test cases by using some AI (Artificial Intelligence) algorithms.

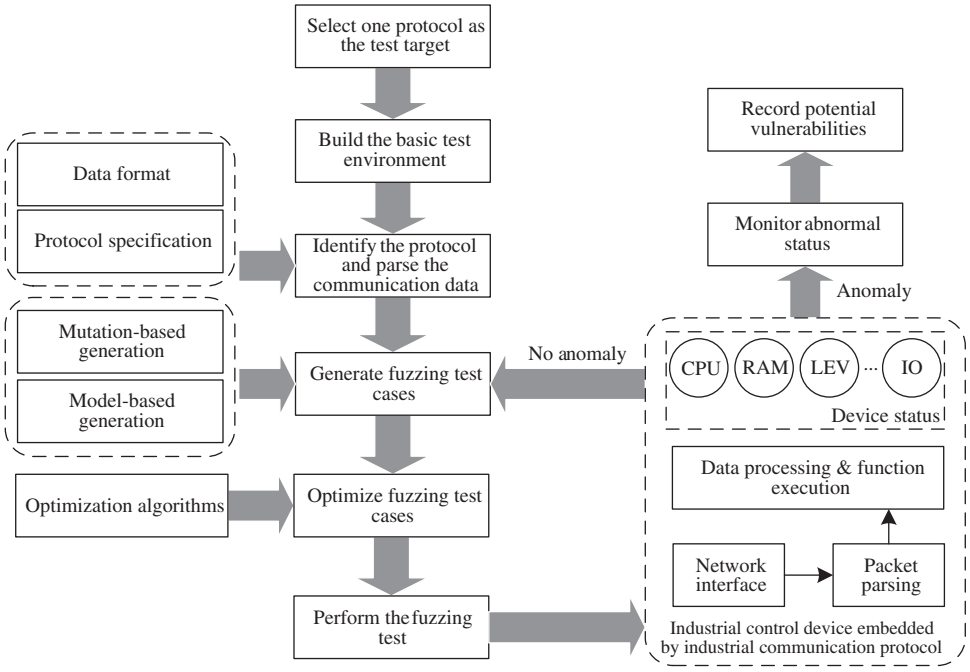

Figure 1: Basic fuzzing test framework for industrial communication protocols

3 Fuzzing Test Case Optimization Approach Based on Improved Genetic Algorithm

Due to its fine global search capability, genetic algorithm can be used to implement the data optimization in the test case generation process [27–29]. Although the test efficiency can be improved to some extent by the data optimization of traditional genetic algorithm, the optimization performance is still less than ideal. In order to further raise the test efficiency, it is necessary to strengthen the optimization performance by improving the traditional genetic algorithm, and the ultimate goal is to exploit more vulnerabilities without changing the number of teat cases.

Distorted data assumption: since the protocol data essentially consist of a series of data fields which are arranged in a certain order. As a result, the protocol data can be expressed as a one-dimensional vector $case = [c_1 c_2 c_3 \cdots c_r]$, here c_r is the corresponding field value in the protocol data. Furthermore, we suppose that the distorted data can be used to exploit certain vulnerability, and conform to the following principle: we can select m ($m \leq r$) fixed fields in the protocol data, and define the corresponding data range in each field. When m field values in one test case fall into the selected range, the protocol data in this test case can be considered as the distorted data.

Fig. 2 shows the main execution process of improved selection genetic algorithm (IS-GA), and the detailed steps are sketched as follows:

Step 1: get 2n original packets of one industrial communication protocol as the initial population;

Step 2: according to the fitness function, calculate each fitness value of $2n$ individuals in the initial population. The basic calculation process is described below:

1) The objective function of similarity F^i_{sim}: calculate each intermediate value of data range for all selected m fields respectively, and obtain the central use case cu, which can be used to calculate F^i_{sim} of

individual i. More specifically, the similarity S_i between individual i and the central use case cu can be further obtained by using Euclidean distance:

$$\begin{cases} F_{sim}^{i} = 1 - S_i \\ S_i = \left(\sum_{k=1}^{m} \left((c_k^i)^2 - (c_k^{cu})^2 \right) \right)^{1/2} \end{cases} \tag{1}$$

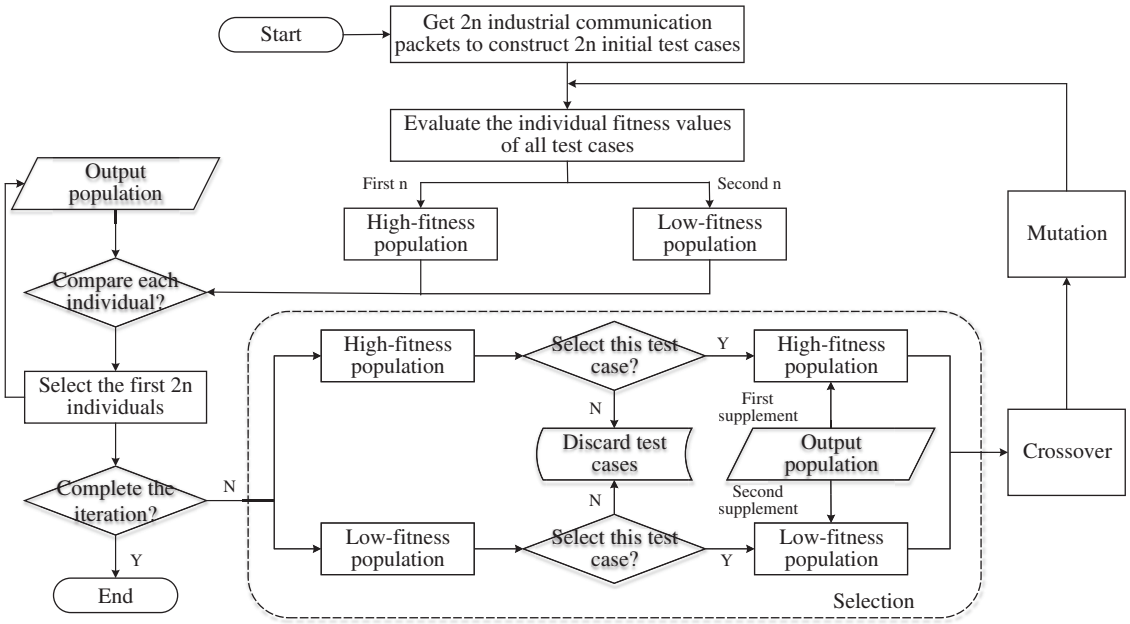

Figure 2: Main execution process of improved selection genetic algorithm

Here, c_k^i represents the k-th field value of individual i, and c_k^{cu} represents the k-th field value of the central use case cu.

2) The objective function of coverage F_{cov}^i: compare m field values of individual i with the data ranges of m fields in the distorted data respectively, and calculate the corresponding coverage value. Actually, the larger the number of fields whose values belong to the data ranges is, the bigger the coverage value is. In other words, the more likely it is that one vulnerability can be successfully exploited. The objective function of coverage F_{cov}^i can be obtained by:

$$F_{cov}^i = f_{num}^i / m \tag{2}$$

Here, f_{num}^i represents the number of fields whose values in individual i are within the corresponding data range in the distorted data.

3) The fitness function F_{fit}^i: when evaluating the individual fitness value, we first analyze whether this individual can successfully exploit some vulnerability, and the basic judgement criterion can be briefly summed up as follows: we compare all m field values of individual i with the data ranges of m fields in the distorted data respectively, and if all field values of individual i remain within the corresponding data range in the distorted data, this individual can be considered to identify one potential vulnerability. So, the fitness value of individual i can be set to 1; otherwise, it is considered that this individual cannot identify any vulnerability. The fitness function F_{fit}^i can be obtained by:

$$F_{fit}^{i} = \alpha F_{sim}^{i} + \beta F_{cov}^{i} \tag{3}$$

Here, we define α and β as the weight coefficients, and $\alpha + \beta = 1$ ($\alpha \in [0, 1]$, $\beta \in [0, 1]$).

Step 3: according to the fitness value of each individual, rearrange $2n$ test cases from largest to smallest. After that, the first n individuals are assigned to the high-fitness population, and the remaining n individuals are naturally divided into the low-fitness population. In the first generation, because there is no individual in the output population, all $2n$ individuals in the high-fitness and low-fitness populations are automatically copied into the output population. Differently, from the second generation, the fitness values of all individuals in the high-fitness and low-fitness populations are compared with the ones in the output population, and the first $2n$ individuals are copied into the output population, while the individuals in the high-fitness and low-fitness populations remain unchanged.

Step 4: judge whether the iteration process of genetic algorithm is completed: if the stop condition is satisfied, the algorithm is stopped; otherwise, go to Step 5

The stop condition is defined as follows:

$$T_{cur} > Iter_{\max} \tag{4}$$

Here, T_{cur} is the current iteration time, and $Iter_{\max}$ is the largest number of iterations which is a pre-determined constant before the algorithm runs.

Step 5: perform the selection operation, the crossover operation and the mutation operation on all individuals in the high-fitness and low-fitness populations in order, and then obtain one new initial population. Then go to Step 2.

1) Selection operation:

A. Selection operation on the high-fitness population:

The individual selection process on the high-fitness population is described as follows: in the first generation, the individual selection criteria $Sel_h(1)$ is set to 0, that is, all individuals of the high-fitness population in the first generation can meet the selection requirements. From the second generation, the individual selection criteria $Sel_h(T_{cur})$ can be calculated by Eq. (5), and the individuals who achieve this criteria can be selected. After that, the high-fitness population can be supplemented by the individuals in the output population according to the fitness value of each individual, so that the number of individuals in the high-fitness population remains the same number n.

$$\begin{cases} Sel_h(T_{cur}) = Sel_h(T_{cur} - 1) + \mu\left(1 + \dfrac{Num_h(T_{cur} - 1) - NP_h}{NP_h}\right) \\ NP_h = n/2 \end{cases} \tag{5}$$

Here, μ is a pre-determined constant which meets $0 \leq 2\mu \cdot Iter_{\max} < 1$. $Num_h(T_{cur} - 1)$ is the number of selected individuals in the high-fitness population when the current iteration time is $T_{cur} - 1$.

B. Selection operation on the low-fitness population:

The individual selection process on the low-fitness population is described as follows: in the first generation, the individual selection criteria $Sel_l(1)$ is set to 0, that is, all individuals of the low-fitness population in the first generation can meet the selection requirements. From the second generation, the individual selection criteria $Sel_l(T_{cur})$ can be calculated by Eq. (6), and if $Sel_l(T_{cur}) > Sel_h(T_{cur})$, then $Sel_l(T_{cur})$ is reset to $Sel_h(T_{cur})$, because the individual selection criteria on the high-fitness population must be larger than the one on the low-fitness population. After that, the individuals who achieve this criteria can be selected, and the rest of individuals are discarded. Finally, apart from these individuals in the output population who have supplemented the high-fitness population, the low-fitness population can be supplemented by other individuals in the output population according to the fitness value of each individual, so that the number of individuals in the low-fitness population remains the same number n.

$$\begin{cases} Sel_l(T_{cur}) = Sel_l(T_{cur}-1) + v\left(1 + \dfrac{Num_l(T_{cur}-1) - NP_l}{NP_l}\right) \\ NP_l = n/2 \end{cases} \tag{6}$$

Here, v is a pre-determined constant which meets $0 \le 2v \cdot Iter_{\max} < 1$. $Num_l(T_{cur}-1)$ is the number of selected individuals in the low-fitness population when the current iteration time is $T_{cur}-1$.

The average value $\overline{Sel(T_{cur})}$ of individual selection criteria is calculated by

$$\overline{Sel(T_{cur})} = (Sel_h(T_{cur}) + Sel_l(T_{cur}))/2 \tag{7}$$

2) Crossover operation:

If one individual cannot satisfy the average value $\overline{Sel(T_{cur})}$ of individual selection criteria, the corresponding crossover probability of this individual should be set to the constant ω, otherwise the crossover probability Pc_i can be calculated by Eq. (8). After that, the crossover operation can be further performed.

$$Pc_i = \xi \cdot \frac{F_{fit}^{\max} - F_{fit}^{i}}{F_{fit}^{\max} - \overline{Sel(T_{cur})}} \tag{8}$$

Here, ξ is a pre-determined parameter, which can adjust the crossover probability Pc_i and avoid the occurrence of premature convergence. $F_{fit}^{\max}$ is the maximum fitness value in the population before the crossover operation, and F_{fit}^{i} is the fitness value of individual i.

3) Mutation operation:

Similarly, if one individual cannot satisfy the average value $\overline{Sel(T_{cur})}$ of individual selection criteria, the corresponding mutation probability of this individual should be set to the constant σ, otherwise the mutation probability Pm_i can be calculated by Eq. (9). After that, the mutation operation can be further performed.

$$Pm_i = \psi \cdot \frac{F_{fit}^{\max} - F_{fit}^{i}}{F_{fit}^{\max} - \overline{Sel(T_{cur})}} \tag{9}$$

Here, ψ is a pre-determined parameter, which can adjust the mutation probability Pm_i and improve the algorithm stability by avoiding the inappropriate random searching. $F_{fit}^{\max}$ is the maximum fitness value in the population before the mutation operation, and F_{fit}^{i} is the fitness value of individual i.

4 Experimental Results and Compared Analysis

In order to verify the superiority of the proposed approach which can further improve the efficiency of fuzzing test, we select Siemens S7 communication protocol as a target object to perform some compared experiments and analysis. In practice, Siemens S7 communication protocol belongs to a kind of special-purpose industrial control protocol, which is based on the typical TCP/IP protocol to perform the real-time control operation and data acquisition between upper computers and Siemens PLCs (Programmable Logic Controllers) [30]. Moreover, Siemens S7 communication protocol is a function/command-oriented industrial Ethernet protocol, whose main communication way is to establish the request/reply connection. That is, if one adversary sends one request packet which not only conforms to the basic protocol specification but also is interspersed with some distorted data, the corresponding PLC may potentially result in a crash because it cannot handle or ignore this exception. The basic message structure of Siemens S7 communication protocol includes the following parts: Header is the S7 application protocol header, which can identify the Siemens S7 application data unit, mainly including the Protocol ID, PDU

Type, Reserved, PUD Reference, Parameter Length, Data Length, Error Class and Error Code; Parameter achieves the setting of read/write function; Data is the specific data during the execution of read/write function. In particular, we select 6 different fields in the basic S7 message structure as the actual testing vectors in the distorted data assumption, and these 6 fields include Reserved, PUD Reference, Parameter Length, Item Count, Variable Specification and Length (the last three fields come from Field Parameter).

In these experiments, we build an industrial communication environment which uses Siemens S7 communication protocol as the basic transmission medium of control operation and data acquisition. Furthermore, we capture lots of S7 communication packets, which are used as the initial data for the test case generation in fuzzing test. Additionally, the initial data are optimized by the proposed approach and traditional genetic algorithm respectively, and our ultimate purpose is to discuss the proposed approach has a better optimization effect for industrial communication data. In other words, when we perform the fuzzing test, the proposed approach can contribute to identifying more potential vulnerabilities under the same number of test cases.

4.1 Optimization Performance Comparison

Firstly, we use a group of S7 communication packets to compare the average and best individual fitness values in each generation between the proposed approach (IS-GA), single selection genetic algorithm (SS-GA) and traditional genetic algorithm (GA). Different from IS-GA, SS-GA only uses the single selection operation on the high-fitness population as the final selection operator to optimize test cases. Tab. 1 gives the basic parameter setting in our experiments, and Fig. 4 depicts the average and best individual fitness curves of three different algorithms.

Table 1: Experimental parameter setting

Parameter	n	m	α	β	μ	v	ξ	ψ
Value	10	6	0.4	0.6	0.005	0.004	0.6	0.5

As shown in Fig. 3, when the number of iterations is set to 100, the average values of GA, SS-GA and IS-GA calculated by the average individual fitness curves are 0.5227, 0.6629 and 0.6781, respectively. Similarly, the average values of GA, SS-GA and IS-GA calculated by the best individual fitness curves are 0.6810, 0.7291 and 0.7647, respectively. From the compared results we can conclude that, the average and maximum individual fitness values of IS-GA are generally larger than the ones of SS-GA and GA. Moreover, these results indirectly indicate that the proposed approach not only has a preferable effect on global optimization, but also enhances the search capability of each individual. As a result, the probability of each test case to successfully identify one potential vulnerability is raised with the enhancing of the optimization capability.

Additionally, in order to explain the universality and stability of optimization performance, we choose 10 groups of different S7 communication packets to perform the data optimization of test cases, and compare the average fitness values of different test cases, including the test cases optimized by GA, the test cases optimized by SS-GA and the test cases optimized by IS-GA. Tab. 2 shows the compared experimental results, and we can see that either average fitness value of test cases optimized by SS-GA or IS-GA is larger than the one of test cases optimized by GA under the same iteration number. That is, both SS-GA and IS-GA have played a positive role in promoting the quality of test cases. Furthermore, compared with the average fitness values of test cases optimized by GA and SS-GA, the one of test cases optimized by IS-GA can be raised by 45.60% and 14.50%. In terms of average experimental results, the proposed approach has a stable and excellent optimization performance, and can further optimize the initial data

under the same iteration number to generate more effective and practicable test cases, which can be used to discover more flaws for industrial communication protocols.

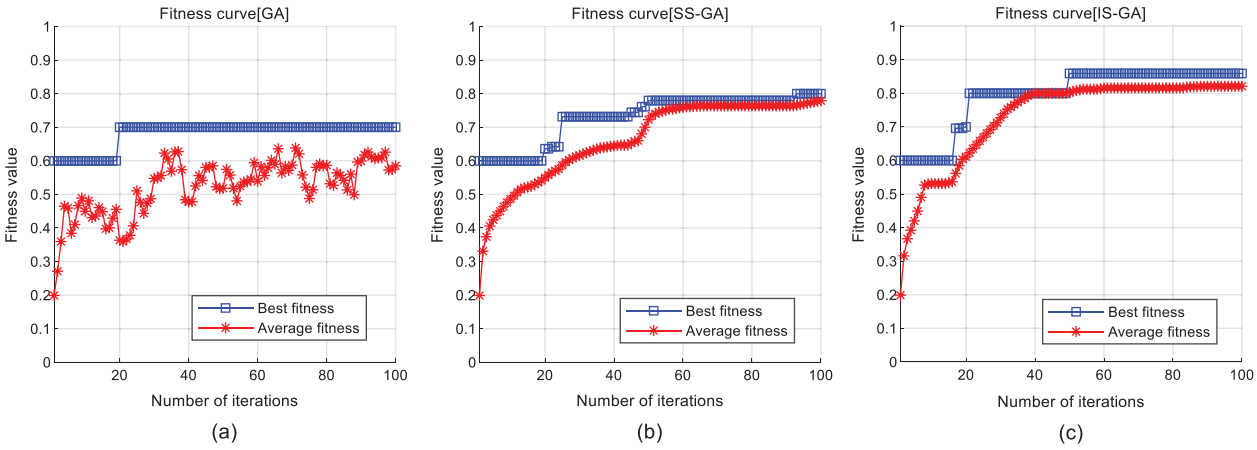

Figure 3: Average and best individual fitness curves of three different algorithms under 100 iterations

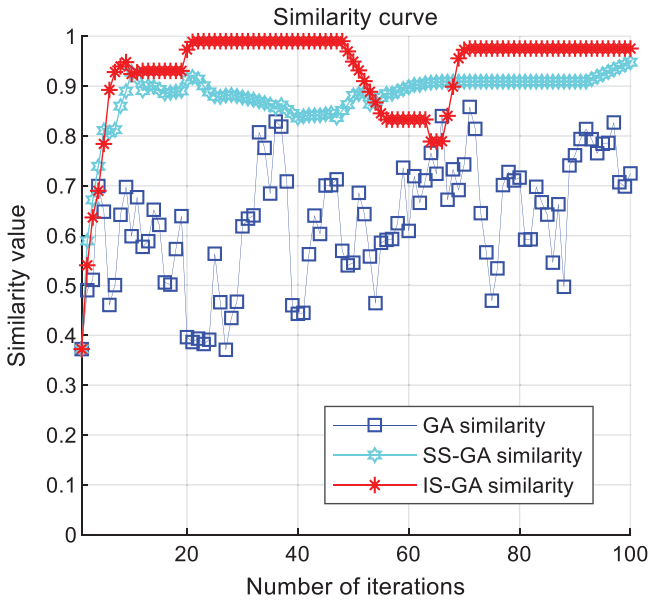

Figure 4: Similarity curves of three different algorithms under 100 iterations

4.2 Similarity Analysis

The similarity can reflect the deviation degree between each individual and the central use case, and the corresponding similarity value can be calculated by Eq. (1) to evaluate the level of similarity. Actually, the closer each individual is to the distorted data, the larger the similarity value seems. In other words, if one individual which represents one test case has a larger similarity value, the possibility to successfully identify one potential vulnerability will become higher. Fig. 4 compares different similarity values of GA, SS-GA and IS-GA under 100 successive iterations. From this figure we can see that the average similarity values of GA, SS-GA and IS-GA are 0.6285, 0.8778 and 0.9305 respectively, and the average similarity value of IS-GA is superior to the ones of GA and SS-GA. Additionally, the maximum

similarity values of GA, SS-GA and IS-GA are 0.8583, 0.9481 and 0.9902 when the iteration number are 71, 100 and 22, respectively. According to the above compared results, it can be concluded that IS-GA can present the fine characteristics of global searching and fast convergence, and the individuals in IS-GA are closer to the distorted data. That is, the test cases optimized by IS-GA hold more sufficient ability in the vulnerability exploiting for industrial communication protocols.

Table 2: Average fitness value comparison of different test cases

	Test cases optimized by GA	Test cases optimized by SS-GA	Test cases optimized by IS-GA
1	0.5581	0.7000	0.8863
2	0.4534	0.6826	0.8256
3	0.5303	0.7000	0.8000
4	0.5621	0.6802	0.8000
5	0.5417	0.7000	0.8000
6	0.5726	0.7000	0.8150
7	0.5826	0.7343	0.8000
8	0.6031	0.7595	0.7813
9	0.5378	0.7213	0.7940
10	0.6214	0.7000	0.8000
average	0.5563	0.7078	0.8102

4.3 Coverage Analysis

The coverage can reflect the usability scale of all test cases in each iteration, and it indirectly indicates that whether one individual can successfully cover the distorted data. As described in the execution process of our approach, the coverage degree can be calculated by Eq. (2) to evaluate the extent of coverage. Similarly, the greater all individuals cover the distorted data, the larger the coverage value seems. That is to say, if one individual which represents one test case has a larger coverage value, the possibility to successfully identify one potential vulnerability will become higher. Fig. 5 shows different coverage value changes of GA, SS-GA and IS-GA under 100 successive iterations. More precisely, the average coverage values of GA, SS-GA and IS-GA are 0.4522, 0.5196 and 0.6033, respectively. Additionally, the maximum coverage values of GA, SS-GA and IS-GA are 0.5000, 0.6667 and 0.7583 when the iteration number are 28, 53 and 88, respectively. From the compared results we can conclude that, the individuals in IS-GA can achieve the higher coverage value, and IS-GA has a fine power to increase the coverage value by enlarging the usability scale of test cases. In brief, IS-GA can indirectly improve the successful chances to exploit one potential vulnerability due to its high coverage extent.

4.4 Influence of ξ and ψ

In effect, the crossover and mutation operations are two significant links in the design of genetic algorithm, which may have a powerful influence on the algorithm performance. In the proposed approach, we introduce two pre-determined parameters ξ and ψ to skillfully adjust the crossover probability and the mutation probability respectively, and our ultimate goal is to effectively improve the optimization performance. Furthermore, the parameter ξ can help to overcome the shortcoming of premature convergence and stagnation, and the parameter ψ can reduce the possibility of inappropriate

random searching to enhance the stability. In order to explain the influence caused by the parameters ξ and ψ, we compare the average and best individual fitness values of IS-GA under different parameters ξ and ψ. To be specific, Fig. 6 depicts the average and best individual fitness curves when the parameter ξ changes, and Fig. 7 depicts the average and best individual fitness curves when the parameter ψ changes. Additionally, Tab. 3 shows all average values calculated by the average individual fitness curves under different ξ when ψ is set to 0.5, and Tab. 4 shows all average values calculated by the best individual fitness curves under different ψ when ξ is set to 0.6.

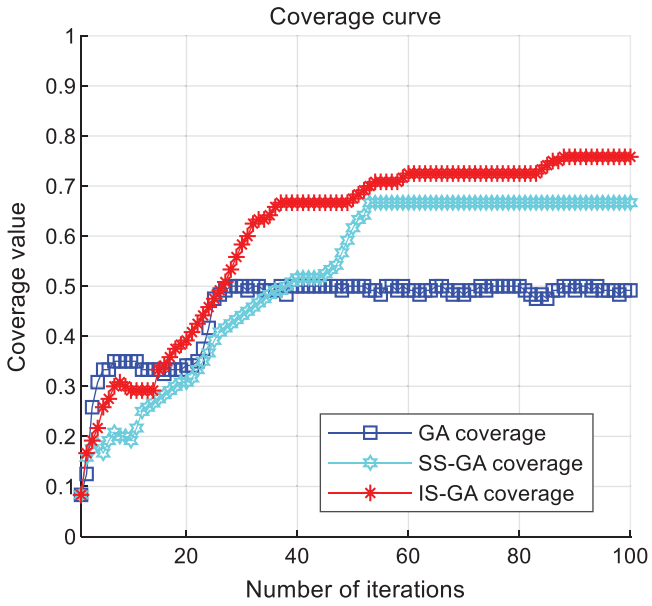

Figure 5: Coverage curves of three different algorithms under 100 iterations

As shown in these figures and tables, with the augment of the parameters ξ and ψ, both of the average values calculated by the average and best individual fitness curves increase first and then decrease. That is to say, by selecting the appropriate parameters ξ and ψ, the fitness value can reach a peak, and IS-GA can achieve the optimal optimization effect. Exactly, the main reasons for this trend can be summarized as follows: on the one hand, when the parameters ξ and ψ gradually increase with certain realms, the crossover and mutation probabilities get larger as well, and the optimization effect of IS-GA can be further enhanced; on the other hand, when the parameters ξ and ψ exceed a certain value respectively, IS-GA can be approximated as one inappropriate random algorithm due to the excessive crossover and mutation probabilities, and the corresponding optimization effect starts to run out. Therefore, we select $\xi = 0.6$ and $\psi = 0.5$ as the relatively desirable parameters in our experiments, which can improve the developmental capability to optimize test cases. In practice, according to different industrial communication protocols and optimization requirements, we can design the most suitable parameters ξ and ψ by performing lots of actual experiments under comprehensive considerations of optimization performance.

To sum up, the main reasons for the superiority of the proposed approach can be analyzed as follows: the proposed approach introduces a novel individual selection strategy, which implements the individual selection process according to the customized individual selection criteria. In the early stage of iteration, due to the large number of qualified individuals, the individual selection criteria grows rapidly, and this growth can effectively accelerate the data optimization process. In the later stage of iteration, the number of individuals who can achieve the selection criteria gradually decreases, and the individual selection criteria grows slowly. Differently, through the supplement of individuals in the output population, the

individual diversity can be effectively enriched, and each individual fitness can continue to increase. Therefore, the test cases optimized by the proposed approach can have a greater chance of identifying more potential vulnerabilities in fuzzing test.

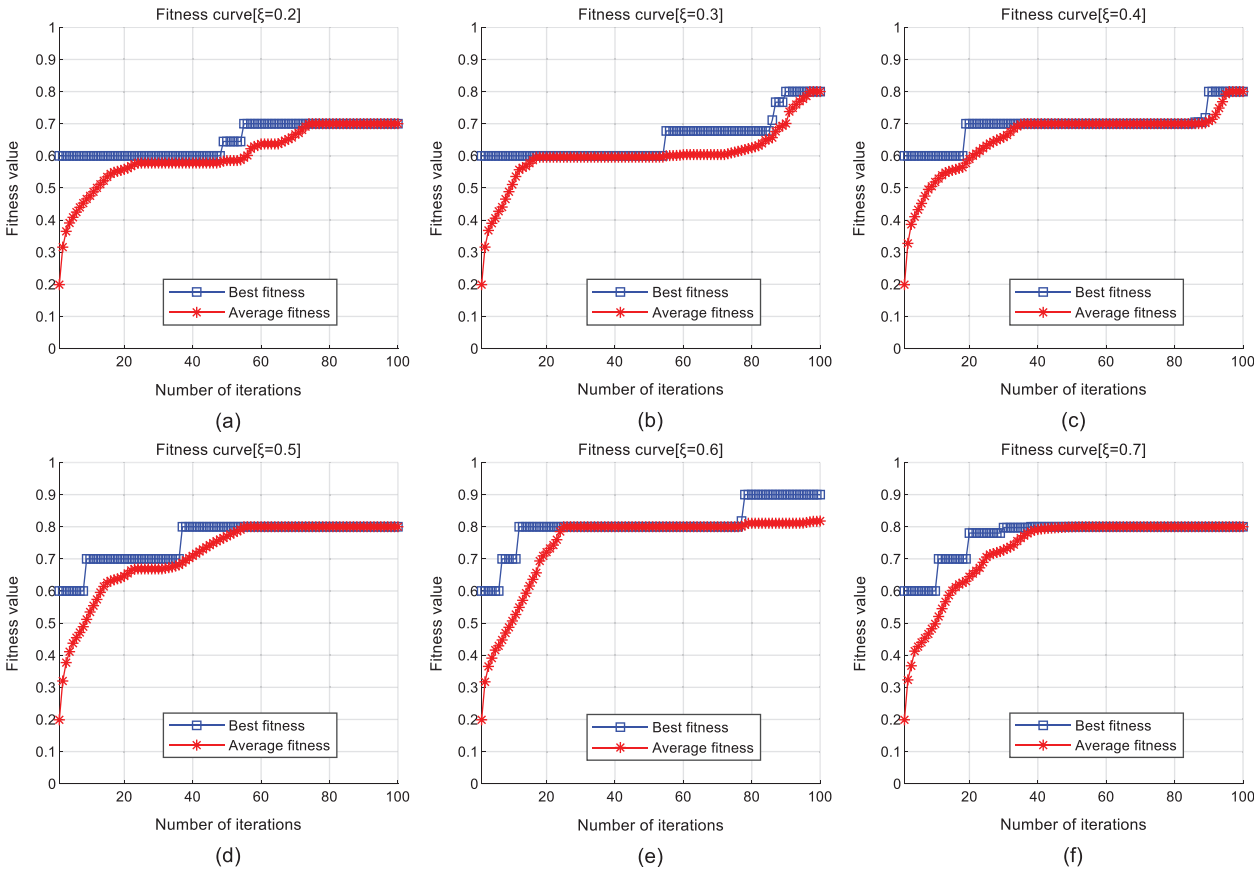

Figure 6: Average and best individual fitness curves under different ξ

Table 3: Average fitness values in average and best individual fitness curves under different ξ

	Average valuein average individual fitness curve	Average valuein best individual fitness curve
$\xi = 0.2$	0.6002	0.6487
$\xi = 0.3$	0.6021	0.6521
$\xi = 0.4$	0.6577	0.6933
$\xi = 0.5$	0.7125	0.7560
$\xi = 0.6$	0.7439	0.8062
$\xi = 0.7$	0.7252	0.7688

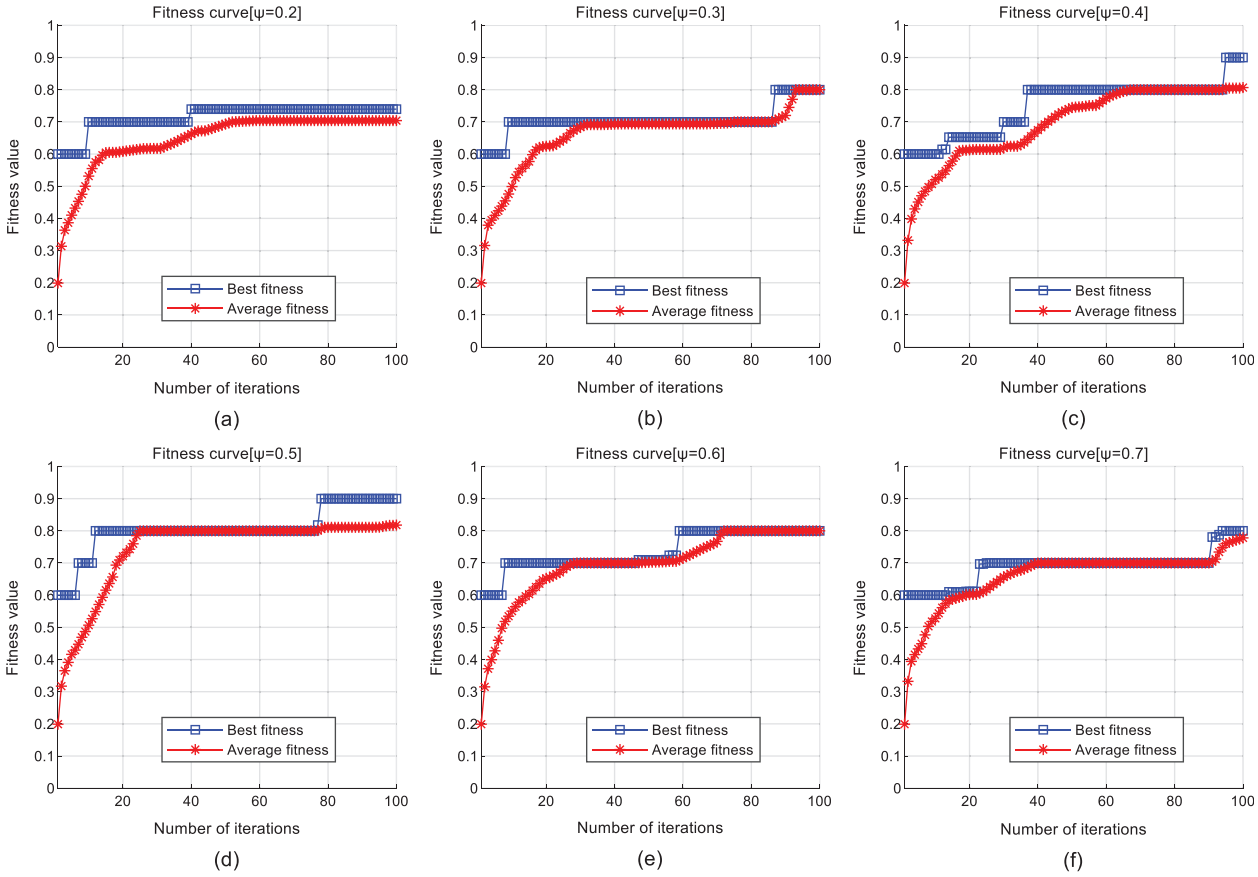

Figure 7: Average and best individual fitness curves under different ψ

Table 4: Average fitness values in average and best individual fitness curves under different ψ

	Average value in average individual fitness curve	Average value in best individual fitness curve
$\psi = 0.2$	0.6451	0.7151
$\psi = 0.3$	0.6603	0.7060
$\psi = 0.4$	0.6931	0.7496
$\psi = 0.5$	0.7439	0.8062
$\psi = 0.6$	0.6967	0.7366
$\psi = 0.7$	0.6565	0.6884

5 Conclusion

In order to improve the efficiency of fuzzing test for industrial communication protocols, this paper focuses on the optimization problem of test cases, and proposes a novel fuzzing test case optimization approach based on improved genetic algorithm. Furthermore, this approach designs a new individual selection strategy for the traditional genetic algorithm to carry out the selection operation, and this strategy can be successfully applied to generate multiple distorted data for different test cases.

In particular, the selection operation based on high and low fitness populations is introduced to enhance the individual selection diversity, which can increase the average fitness value of individuals and further improve the efficiency of test cases. Additionally, we only discuss the optimization effect by theoretic analysis and experimental assumption, and all experimental results demonstrate that this approach can enhance the efficiency and accuracy of test cases in Siemens S7 fuzzing vulnerability exploiting. In the future research, we can further evaluate the proposed approach in real-world applications of fuzzing test.

Acknowledgement: The authors are grateful to the anonymous referees for their insightful comments and suggestions.

Funding Statement: This work is supported by the Program of Hainan Association for Science and Technology Plans to Youth R & D Innovation (Grant No. QCXM201910), the Natural Science Foundation of Liaoning Province (Grant No. 2019-MS-149), the National Natural Science Foundation of China (Grant Nos. 61802092 and 51704138), the Key Scientific Research Project of Liaoning Provincial Department of Education (Grant No. LZD202002), and the intelligent Manufacturing Standardization and Test Verification Project "Time Sensitive Network (TSN) and Object Linking and Embedding Unified Architecture for Industrial Control OPC UA Fusion Key Technology Standard Research and Test Verification" project, Ministry of Industry and Information Technology of the People's Republic of China.

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

References

[1] S. K. Kim, M. Koppen, A. K. Bashir and Y. Jin, "Advanced ict and iot technologies for the fourth industrial revolution," *Intelligent Automation & Soft Computing*, vol. 26, no. 1, pp. 83–85, 2020.

[2] M. Wan, J. Li, Y. Liu, J. Zhao and J. Wang, "Characteristic insights on industrial cyber security and popular defense mechanisms," *China Communications*, vol. 18, no. 1, pp. 130–150, 2021.

[3] M. Pogliani, D. Quarta, M. Polino, M. Vittone, F. Maggi *et al.,* "Security of controlled manufacturing systems in the connected factory: The case of industrial robots," *Journal of Computer Virology and Hacking Techniques*, vol. 15, no. 3, pp. 161–175, 2019.

[4] J. Lee, H. Choi, J. Kim, J. Kim, D. Jung *et al.,* "Identifying and verifying vulnerabilities through PLC network protocol and memory structure analysis," *Computers, Materials & Continua*, vol. 65, no. 1, pp. 53–67, 2020.

[5] T. Gebremichael, L. P. I. Ledwaba, M. H. Eldefrawy, G. P. Hancke, N. Pereira *et al.,* "Security and privacy in the industrial internet of things: Current standards and future challenges," *IEEE Access*, vol. 8, pp. 152351–152366, 2020.

[6] X. Pan, Z. Wang and Y. Sun, "Review of PLC security issues in industrial control system," *Journal of Cyber Security*, vol. 2, no. 2, pp. 69–83, 2020.

[7] S. Vitturi, C. Zunino and T. Sauter, "Industrial communication systems and their future challenges: next-generation Ethernet, IIoT, and 5G," *Proc. of the IEEE*, vol. 107, no. 6, pp. 944–961, 2019.

[8] T. Cruz, L. Rosa, J. Proenca, L. Maglaras, M. Aubigny *et al.,* "A cybersecurity detection framework for supervisory control and data acquisition systems," *IEEE Trans. on Industrial Informatics*, vol. 12, no. 6, pp. 2236–2246, 2016.

[9] Z. Darias, A. Serhrouchni and O. Vogel, "Taxonomy of attacks on industrial controls protocols," in *Proc. 2015 Int. Conf. on Protocol Engineering and New Technologies of Distributed Systems*, Paris, France, pp. 1–6, 2015.

[10] T. Vollmer and M. Manic, "Cyber-physical system security with deceptive virtual hosts for industrial control networks," *IEEE Trans. on Industrial Informatics*, vol. 10, no. 2, pp. 1337–1347, 2014.

[11] S. Lee, S. Lee, H. Yoo, S. Kwon and T. Shon, "Design and implementation of cybersecurity testbed for industrial IoT systems," *Journal of Supercomputing*, vol. 74, no. 9, pp. 4506–4520, 2018.

[12] Z. Li, H. Zhao, J. Shi, Y. Huang and J. Xiong, "An intelligent fuzzing data generation method based on deep adversarial learning," *IEEE Access*, vol. 7, pp. 49327–49340, 2019.

[13] W. Xu, Y. Tao, C. Yang and H. Chen, "MSICST: Multiple-scenario industrial control system testbed for security research," *Computers, Materials & Continua*, vol. 60, no. 2, pp. 691–705, 2019.

[14] W. Su, A. Antoniou and C. Eagle, "Cyber security of industrial communication protocols," in *Proc. 2017 22nd IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA)*, Limassol, Cyprus, pp. 1–4, 2017.

[15] L. Rosa, M. Freitas, S. Mazo, E. Monteiro, T. Cruz *et al.,* "A comprehensive security analysis of a SCADA protocol: from OSINT to mitigation," *IEEE Access*, vol. 7, pp. 42156–42168, 2019.

[16] O. N. Nyasore, P. Zavarsky, B. Swar, R. Naiyeju and S. Dabra, "Deep packet inspection in industrial automation control system to mitigate attacks exploiting Modbus/TCP vulnerabilities," in *Proc. 2020 IEEE Int. Conf. on Big Data Security on Cloud (BigDataSecurity), High Performance and Smart Computing (HPSC) and Intelligent Data and Security (IDS)*, Baltimore, USA, pp. 241–245, 2020.

[17] J. Francois, A. Lahmadi, V. Giannini, D. Cupif, F. Beck *et al.,* "Optimizing internet scanning for assessing industrial systems exposure," in *Proc. 2016 Int. Wireless Communications and Mobile Computing Conf. (IWCMC)*, Paphos, Cyprus, pp. 516–522, 2016.

[18] A. Walz and A. Sikora, "Exploiting dissent: Towards fuzzing-based differential black-box testing of TLS implementations," *IEEE Trans. on Dependable and Secure Computing*, vol. 17, no. 2, pp. 278–291, 2020.

[19] J. Li, B. Zhao and C. Zhang, "Fuzzing: a survey," *Cybersecurity*, vol. 1, no. 6, pp. 1–13, 2018.

[20] H. Liang, X. Pei, X. Jia, W. Shen and J. Zhang, "Fuzzing: State of the art," *IEEE Trans. on Reliability*, vol. 67, no. 3, pp. 1199–1218, 2018.

[21] H. Zhao, Z. Li, H. Wei, J. Shi and Y. Huang, "SeqFuzzer: An industrial protocol fuzzing framework from a deep learning perspective," in *Proc. 2019 12th IEEE Conf. on Software Testing, Validation and Verification (ICST)*, Xi'an, China, pp. 59–67, 2019.

[22] Z. Luo, F. Zuo, Y. Shen, X. Jiao and W. Chang , "Chang etal, ICS protocol fuzzing: coverage guided packet crack and generation," in *Proc. 2020 57th ACM/IEEE Design Automation Conf. (DAC)*, San Francisco, USA, pp. 1–6, 2020.

[23] A. I. Pechenkin and A. V. Nikolskiy, "Architecture of a scalable system of fuzzing network protocols on a multiprocessor cluster," *Automatic Control and Computer Sciences*, vol. 49, no. 8, pp. 758–765, 2015.

[24] S. J. Kim and T. Shon, "Field classification-based novel fuzzing case generation for ICS protocols," *Journal of Supercomputing*, vol. 74, no. 9, pp. 4434–4450, 2018.

[25] R. Ma, D. Wang, C. Hu, W. Ji and J. Xue, "Test data generation for stateful network protocol fuzzing using a rule-based state machine," *Tsinghua Science and Technology*, vol. 21, no. 3, pp. 352–360, 2016.

[26] G. Lemieux and K. Sen, "FairFuzz: A targeted mutation strategy for increasing greybox fuzz testing coverage," in *Proc. 2018 33rd IEEE/ACM Int. Conf. on Automated Software Engineering (ASE)*, Montpellier, France, pp. 475–485, 2018.

[27] B. Wu, L. Yun, X. Jin, B. Liu and G. Wei, "Study on the fuzzing test method for industrial supervisory control configuration software based on genetic algorithm," in *Proc. 2016 11th Int. Conf. on Reliability, Maintainability and Safety (ICRMS)*, Hangzhou, China, pp. 1–6, 2016.

[28] M. B. Bashir and A. Nadeem, "Improved genetic algorithm to reduce mutation testing cost," *IEEE Access*, vol. 5, pp. 3657–3674, 2017.

[29] Z. Y. Wei, J. Q. Wang, X. Q. Shen and Q. Luo, "Smart contract fuzzing based on taint analysis and genetic algorithm," *Journal of Quantum Computing*, vol. 2, no. 1, pp. 11–24, 2020.

[30] F. Xiao, E. Chen and Q. Xu, "S7commTrace: A high interactive honeypot for industrial control system based on S7 protocol," in *Proc. 2017 19th Int. Conf. on Information and Communications Security*, Beijing China, pp. 412–423, 2017.