International Journal of Technology Research and Management
ISSN (Online): 2348-9006
Vol 12 Issue 03 March 2025

Generative Adversarial Networks based Vulnerability Detection Model for Digital Twin in Industrial IoT

Nikhil Chaurasia[1], Dr. Pritaj Yadav[2], Dr. Sanjeev kumar Gupta[3]
Research Scholar[1], Assistant Professor[2], Professor[3]
Department of Computer Science & Engineering Ravindra Nath Tagore University, Bhopal[1, 2, 3]
nikhilsub97@gmail.com[1], pritaj.yadav@aisectuniversity.ac.in[2], sanjeevgupta73@yahoo.com[3]

Abstract: *The profound integration of informatisation and industrialization is amplifying security concerns regarding digital twins within industrial IoT (IIoT) network protocols. Current techniques for detecting vulnerabilities in network protocols, primarily based on feature mutation and fuzz testing, encounter drawbacks including reliance on expert experience and incapacity to address unknown protocols. This work concentrates on the automated analysis and formulation of vulnerability detection algorithms to tackle the issues of vulnerability detection for digital twins in IIoT protocols. A method for detecting network protocol vulnerabilities is proposed, utilising a combination of Generative Adversarial Networks (GANs) and mutation algorithms. A network protocol analysis model utilising GANs is employed to thoroughly extract information from message sequences, delineate message formats and associated attributes, and ascertain the structure of the network protocol. Subsequently, an iterative mutation technique, informed by a mutation operator library, is employed to develop targeted test case generation rules, therefore reducing the time necessary to identify vulnerabilities. An automated vulnerability identification approach for unidentified industrial control network protocols has been developed, addressing the present need for automated protocol vulnerability detection in the industrial control sector. The proposed methodology entailed the examination of two industrial control protocols (Modbus TCP and S7), assessing the efficacy of generated cases, vulnerability detection proficiency, test case creation duration, and case diversity. Experimental findings demonstrate that the proposed method achieves a TA indication of up to 89.4%. In the ModbusSlave simulation system, the AD indicator attained 6.87%, markedly decreasing the time needed to produce effective cases and enhancing the efficiency of vulnerability detection in industrial control protocols*.

Keywords: Digital Twin, Generative Adversarial Networks, DMGAN, Vulnerability Detection, Industrial IoT, fuzz testing.

1. INTRODUCTION

An industrial control protocol refers to the rules and agreements that both entities must follow to complete communication or services in adigital twin within industrial control environment. In recent years, with the widespread use of industrial control products and systems, many network security incidents have occurred. Notorious examples include the Stuxnet virus [1] and the global ransomware outbreak [2], both of which caused significant losses to society. At the same time, malicious attackers have exploited vulnerabilities in digital twin within industrial IoT based control networks to launch remote attacks on devices connected to the internet [3], directly affecting the safety of the entire internet. As a result, industrial IoT security issues have garnered widespread attention worldwide.

Vulnerability mining refers to the process of using various technologies and tools to discover security vulnerabilities in

software programs, network protocols, etc., as much as possible. According to statistical investigations, traditional vulnerability mining methods for digital twin within industrial IoT based control network protocols mainly employ reverse engineering, penetration testing, and fuzz testing techniques. Among these, fuzz testing typically involves constructing effective test cases based on the software or protocol specifications being tested and using malicious inputs to cause crashes or abnormal behaviour to discover vulnerabilities in the software or protocol [4]. These methods are primarily applied to known industrial IoT based control network protocols. However, due to high investment costs, long execution times, susceptibility to errors, lack of targeting, poor portability, and low detection efficiency, these methods struggle to achieve intelligent and efficient vulnerability mining.

Therefore, this paper offers a mutation-based approach to mining network protocol vulnerabilities that is based on generative adversarial networks, aiming to solve the current challenges encountered in digital twin within industrial IoT based control network protocol vulnerability mining. The use of generative adversarial networks can reduce the subjectivity in manually analysing protocol construction test cases while lowering the time and labour costs associated with constructing fuzz test cases, thus improving the overall efficiency of the fuzz testing process. By designing guided mutation strategies and methods based on the characteristics of digital twin within industrial IoT based control protocol message fields and the known features of existing vulnerability rules, we can efficiently and quickly guide the generation of effective test cases within a controlled range, achieving more precise and efficient vulnerability mining. In this paper, an improved multi-generator generative adversarial networks (DMGAN) are used to generate test cases, combined with offline fast mutation strategies to address the challenges of vulnerability mining in digital twin within industrial IoT based control network protocols. Finally, by conducting simulation experiments on the Modbus TCP and S7 protocols, the proposed algorithm is validated in terms of performance metrics such as test case diversity, system acceptance rates, and the frequency of triggering faults during fuzz testing. The experimental results demonstrate that the DMGAN model can effectively reduce the time and cost of manual analysis, minimize the uncertainties associated with manual input of test cases and manual analysis without compromising vulnerability detection effectiveness, and enable mutation strategies to guide the model in finding effective test cases more quickly.

In summary, the contributions of this paper are threefold:

1. One approach to mining vulnerabilities in network protocols that makes use of the DMGAN model is proposed, which learns the message format from industrial IoT based control sample data, reducing the manual cost and time involved in testing industrial IoT based control systems and improving testing effectiveness.
2. Based on the characteristics of industrial IoT based control protocol message fields and the principles of existing vulnerabilities, a method using mutation operator libraries and packet rules is proposed. This method aims to design guided offline fast mutation strategies to guide the generation of effective test cases, thereby discovering effective cases more quickly.
3. An iterative feedback model is used to continuously filter and guide offline mutations of test cases. The fuzz testing results are combined to further guide mutation strategies and methods. The mutated cases are then fed into the model for training and testing, achieving deeper and broader testing to uncover more potential vulnerabilities.

Section 2, overview of previous research on vulnerability mining using GANs, mutations, and in section 3, proposed combination of GANs and mutation strategies for vulnerability discovery. In section 4, details experimental setup, datasets, performance evaluation of proposed approach, and in section 5, summarizes results, improvements, and future directions for vulnerability mining.

2. RELATED WORK

2.1 Deep Learning and Generative Adversarial Networks

In recent years, machine learning and deep learning have made significant breakthroughs in technology. Particularly, the development of deep learning has enabled computers to possess powerful perception abilities. At the same time, deep learning has attracted attention from the tech industry and shown immense potential for applications. In fields such as gaming, robotics, machine translation, speech recognition, autonomous driving, navigation, intrusion detection, multi-agent collaboration, and recommendation systems, deep learning has achieved performance comparable to, or even surpassing, that of humans.

As technology rapidly advances, researchers have shifted their focus from machine perception to machine creation, using generative techniques in machine learning to enable machines to create new things. The birth of generative adversarial networks has reshaped our understanding of

traditional generative models and has already achieved remarkable results. Adversarial networks represent a new milestone in the field of artificial intelligence. The "father of GANs," Goodfellow, who studied under deep learning master Yoshua Bengio at the University of Montreal, was inspired by biologist Leigh Van Valen's "Red Queen Hypothesis" and conceived the idea of adversarial networks. After multiple attempts, he achieved excellent results in generating images using GANs [5]. However, compared to other deep learning networks, the training of generative adversarial networks is unstable, and issues such as non-convergence, vanishing gradients, and mode collapse can occur [6]. Therefore, this paper proposes an improved generative adversarial networks model to overcome the challenges of non-convergence, vanishing gradients, mode collapse, and poor diversity in generated data during training.

2.2 Fuzz Testing

Fuzz testing has developed over more than 20 years into a widely used vulnerability discovery technique [7]. In 1989, Professor Barton Miller from the University of Wisconsin-Madison introduced the concept of fuzz testing and tested the robustness of applications under the UNIX system [8]. Since then, more and more researchers have focused on fuzz testing, proposing various new ideas and methods. For example, Porter et al. proposed the PROTOS [9] test suite, which generates structured test data using protocol specifications and was the first to apply fuzz testing to network protocol testing. Later, Aitel developed the first custom fuzzing framework, SPIKE [10], which was subsequently improved by other researchers. Peach [11] was used for file fuzz testing and became popular among security testers due to its easy-to-understand scripting language and cross-platform testing capability. AFL [12], developed by security researchers, is a coverage-guided fuzzing tool that has significantly influenced fuzz testing technology and continues to be improved. Recent research has attempted to combine deep learning with AFL, using sequence-to-sequence neural network models to enhance the effectiveness of the AFL fuzzer [13]. Although fuzz testing has proven effective in vulnerability discovery, the randomness in generating test cases and the complexity of the mutation process remain significant challenges. As a result, finding vulnerabilities requires substantial time and computational resources. The data generation process in fuzz testing is usually based on specific rules and algorithms [14], making it difficult to generate complex input data that exists in the real world, and thus, hard to uncover potential vulnerabilities. Moreover, current fuzz testing techniques heavily rely on the experience of security experts.

In past research, deep learning methods were typically used as auxiliary tools, but recent studies have begun applying them as core methods in fuzz testing for industrial IoT network protocols. One example is the proposal by Lv et al. [15] to create valuable binary seed files by means of machine learning. In their recommendation for determining input-specific mutation procedures, Böttinger et al. [16] proposed Q-learning algorithms. In their study, Godefroid et al. [17] investigated methods for learning the grammar of non-binary PDF data objects using neural network-based learning approaches. These research, from different perspectives, have enhanced the application of machine learning in fuzz testing [18]. Nevertheless, fuzz testing has mainly been applied to known network protocols, making it less suitable for unknown domains. This limitation constrains the broader application and development of fuzz testing.

Therefore, this paper uses an improved generative adversarial networks model to conduct deep exploration and analysis of message structures, generating a large number of test cases. Based on an analysis of protocol message fields and the principles of vulnerability triggering, the paper proposes a mutation strategy and method based on a mutation operator library and packet rules to guide the generation of effective test cases. Finally, by analyzing the test results and dynamically adjusting the mutation strategy, the proposed method conducts fuzz testing on industrial control protocols, achieving more extensive and in-depth testing, thereby uncovering more potential vulnerabilities. The combination of generative adversarial networks models and mutation strategies integrates generative-based and mutation-based fuzz testing techniques, not only recognizing unknown network protocol structures and expanding the scope of application, but also reducing manual and time costs, while shortening the time required to discover vulnerabilities.

3. AN APPROACH TO VULNERABILITY MINING UTILIZING GANS AND MUTATION STRATEGIES IN COMBINATION

3.1 Overall Architecture

This paper adopts a approach to vulnerability mining that use GANs in conjunction with mutation techniques to address the current challenges in network protocol vulnerability mining. The method focuses on improving test case diversity, system acceptance rates, and the number of triggered vulnerabilities, aiming to shorten the time required to generate test cases and improve fuzz testing results. By training the improved generative adversarial networks, this

method effectively reduces the time and cost of manual analysis, eliminates the lack of objectivity in manual testing, and enhances the diversity of test cases. Additionally, to efficiently identify effective cases, this paper introduces guided mutation strategies and methods based on known vulnerability characteristics and protocol format features.

The overall architecture of the method involves data pre-processing, generative model design, mutation strategy construction, and result analysis and feedback, as shown in Figure 1.

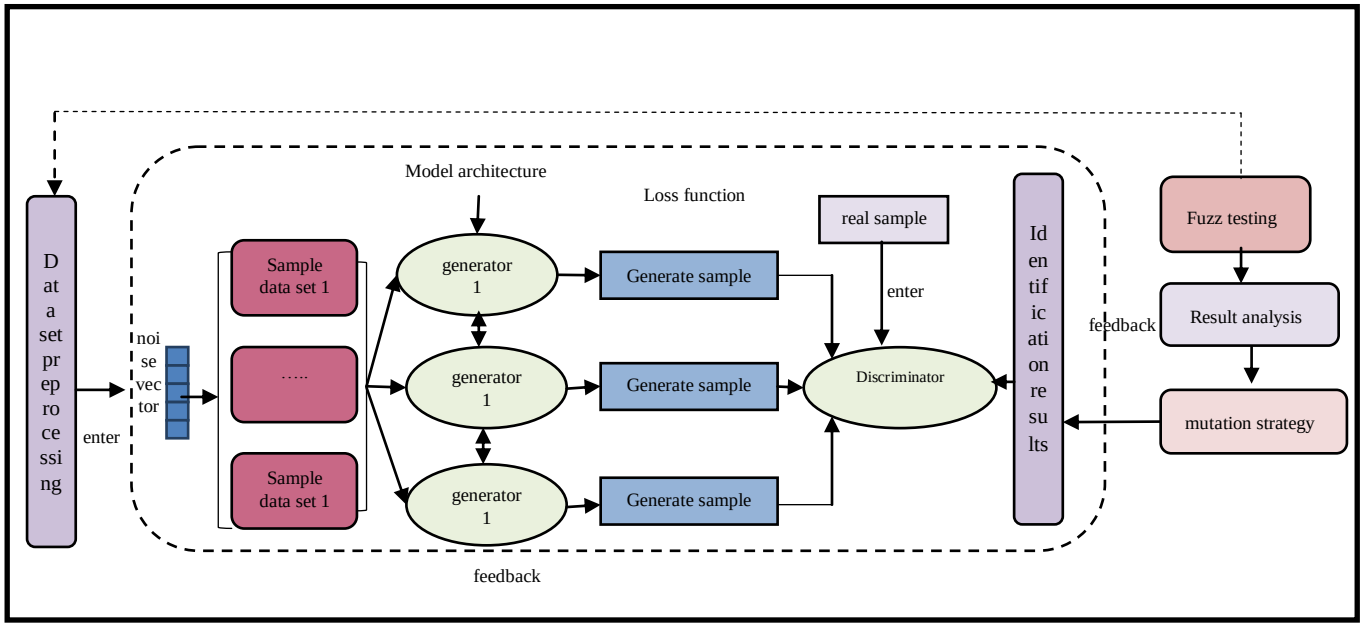

Figure 1: Overall framework of the proposed method

First, the initial dataset of messages is pre-processed, and the processed data is input into the model for training and optimization to automatically generate test cases similar in structure to real-world data. Then, by combining fuzz testing and the analysis of test results, effective anomaly information is obtained to identify valid test cases. The anomaly information mainly includes abnormal responses during fuzz testing and content from an offline vulnerability database. Finally, the test results are fed back into the data processing and model training process. Based on this feedback, the mutation strategy is adjusted, optimizing the mutation operations on the test cases.

3.2 Data Pre-processing Based on Image Format Conversion

To further enhance the training process, the data pre-processing workflow must ensure that the message data is converted into image data in the target format. This is achieved by performing data cleaning [19], base conversion [20], data frame alignment [21], data clustering [22], and format conversion on the captured communication message dataset [23], converting the message data into image data for model training to facilitate deep learning analysis of message formats [24].

In an industrial IoT control network communication environment, data packets exist in the form of sequences, consisting of a message header and a data field [25]. The message header contains protocol control information used for managing data transmission and processing, and it follows a relatively fixed format [26]. The data field contains the actual transmission data from the application layer, and since the instruction information varies, the length and content of the data field differ. It is important to note that communication data frames within the same protocol cluster generally follow a similar and fixed protocol format [27].

The primary goal of the pre-processing stage is to clean, align, convert, and reformat the data frames, transforming the data into an image format for model training. The original data frame sequence is formally represented as $T_{1:n} = (f_1, f_2, f_3, \cdots, f_x, \cdots, f_n), f_x \in E, T_{1:n} \in T^*$. Here, F represents the set of hexadecimal data composed of letters and numbers, and S∗S^*S∗ represents the set of data frame sequences [28]. The base conversion process converts hexadecimal protocol

data frames into decimal format and stores them in specific files. After standardizing the data format for the input into the training model, the data in the file undergoes further format conversion. The one-dimensional vector is converted into a three-dimensional array with 3 rows, 32 columns, and 32 channels, where "3" represents the red, green, and blue (RGB) channels, and "32" represents the height and width of the array, i.e., a $32 * 32$ pixel image.

The data processing flow is shown in Figure 2. During the data alignment operation, zero-padding is introduced, so many pixels in the image will have an RGB value of zero, resulting in a fairly monotone color scheme for the training images [29].

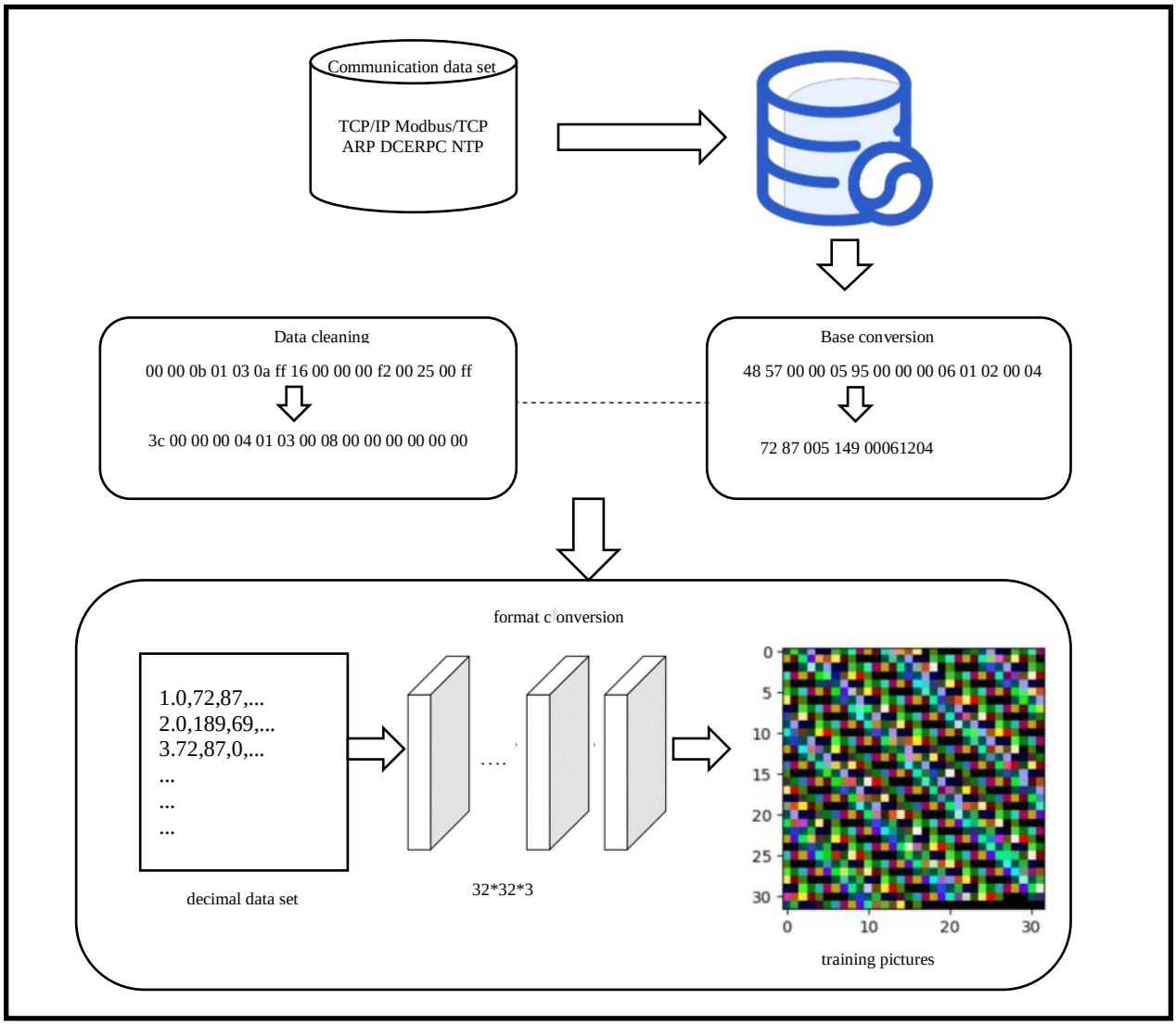

Figure 2: Data pre-processing

3.3 Improved Generative Model Design

The improved generative adversarial network (GAN) model based on deep learning is designed to learn the format and structure of network protocol messages[30]. It generates diverse and differentiated test data packets [31], enhancing the model's generalization capability and robustness. During the model design process, training, parameter tuning, and iterative optimization are comprehensively considered. The

goal is to maintain a balance between the variability and similarity of generated samples to real-world data while ensuring sample diversity.

The principles of the improved generative adversarial networks model (DMGAN) proposed in this paper are as follows: The DMGAN model introduces a Gaussian noise vector and applies a Gaussian Mixture Model (GMM) to parameterize the latent space vector [32]. The parameterized latent space vector is then reparametrized to produce diverse, back-propagation-friendly sample data. Samples are drawn from a specified Gaussian distribution and input into the generator, where multi-modal generators are used to learn the spatial structure of the input data packets. Finally, a discriminator evaluates the authenticity of the generated data, and training ends when the results reach a Nash equilibrium. In this context, Nash equilibrium refers to the optimal state where the generator and discriminator are perfectly balanced[33], meaning the generator's samples can no longer be distinguished from real data by the discriminator. This state is dynamic and is continually updated and adjusted based on past performance.

GMM enhances the modelling capability of prior distributions and the diversity of generated samples without increasing the model's depth. By introducing a mechanism for diversity, the model's generative capabilities are strengthened, allowing it to produce varied samples even with limited data. The Gaussian distribution is defined as follows:

$$g(x \mid \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \mathrm{f}^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (1)$$

Define the distribution of A as a mixture of Gaussians.

$$q_A = \sum_{i=1}^{N} \frac{h(A|\mu_i, \Sigma_i)}{N} \quad (2)$$

This represents the probability of sampling A from the Gaussian distribution $N(\mu_i, \Sigma_i)$, where each Gaussian distribution has two parameters: $h(A \mid \mu_i, \Sigma_i)$and σ_i.Multiple Gaussian distributions are weighted and mixed according to a specified ratio to obtain a new probability distribution.

Since $A$is a non-differentiable random variable, the gradients of the two parameters in the above model cannot be directly back propagated through the sample parameter A. Therefore, a reparameterization technique is introduced. The principle is to transform A into a differentiable variable by standardizing each Gaussian distribution, as shown below:

$$A = \mu_i' + \sigma_i' \epsilon, \epsilon \sim N(0,1) \quad (3)$$

In the above, μ_i' and σ_i' are the reparameterized, standardized mean and variance, respectively, and represent auxiliary noise variables drawn from a standard normal distribution. This transforms Ainto a differentiable variable. Given multiple Gaussian mixture distributions $\boldsymbol{q}_A^i$, the Gaussian noise vector $A$is defined as:

$$\boldsymbol{q}_A^i = \sum_{j=1}^{N} \frac{h\left(\left(\mu_i + \sigma_i \epsilon | \mu_j^i, \Sigma_j^i\right)\right)}{N} \quad (4)$$

where μ_j^i represents the mean of the i Gaussian mixture distribution, and Σ_j^irepresents its variance.

The multi-mode generator learns the spatial structure of the input data sequence, generating test cases. The loss function of the model is composed of both global and local losses. The following equation defines the optimized loss function for model training:

$$\min_{H_{1,K},D} \max_{E} J(H_{1:K}, D, E) = E_X \sim Q_{\text{data}}[\log E(X)] + E_X \sim Q_{\text{modfl}}[\log(1 - E(X))] - \beta\{\sum_{k=1}^{K} \pi_k E_X \sim Q_{\mathrm{G}_k}[\log D_k(X)]\} \quad (5)$$

In this formula, the first two terms calculate the loss between multiple generators and discriminators, Q_{data} represents the real data distribution, and Q_{G} represents the generated data distribution [34], while $E$is the discriminator's result. The losses from the multiple generators are computed with the discriminator and summed to obtain the overall loss between generators and the discriminator, expressed as:

$$E(X) = \frac{Q_{\text{data}}(x)}{Q_{\text{data}}(x) + Q_{H_{1,k}}(x)} \quad (6)$$

The final term of the loss function calculates the losses between multiple generators, expressed as:

$$L(H_{1:K}) = F_X \sim Q_{\text{data}}\left[\log \frac{Q_{\text{data}}(x)}{Q_{\text{data}}(x) + Q_{\text{model}}(x)}\right] + F_X \sim Q_{\text{model}}\left[\log \frac{Q_{\text{model}}(x)}{Q_{\text{data}}(x) + Q_{\text{model}}(x)}\right] - \beta\{\sum_{k=1}^{K} \pi_k F_X \sim Q_{\mathrm{H}_k}\left[\log \sum_{j=1}^{k} \pi_j Q_H j(x)\right]\} \quad (7)$$

The DMGAN model used in this paper employs multiple mode generators and discriminators. Samples are randomly drawn from the reparametrized Gaussian mixture model and input into the multi-mode generator for training. The Gaussian mixture model defines $\boldsymbol{\mu} = [\mu_1, \mu_2, \cdots, \mu_N]^{\mathrm{T}}$ and $\boldsymbol{\sigma} = [\sigma_1, \sigma_2, \cdots, \sigma_N]^{\mathrm{T}}$, with a simplified setup that includes diagonal covariance matrices for each component and equal-weighted mixture components. These settings limit the model's ability to approximate more complex distributions. By continuously optimizing and adjusting the parameters of the Gaussian mixture model, as well as the proportions and

weights of these parameters, the model better fits the latent data distribution, maximizing the probability of generating realistic data $Q_{\text{data}}(H(\mu_i{}' + \sigma_i{}'\epsilon)|_\epsilon)$.The structure of the DMGAN model is shown in Figure 3. By combining the Gaussian mixture model with a multi-generator mode, the model can fit more complex sample distributions, ensuring the diversity and validity of generated data while reducing the likelihood of problems such as mode collapse and convergence difficulties.

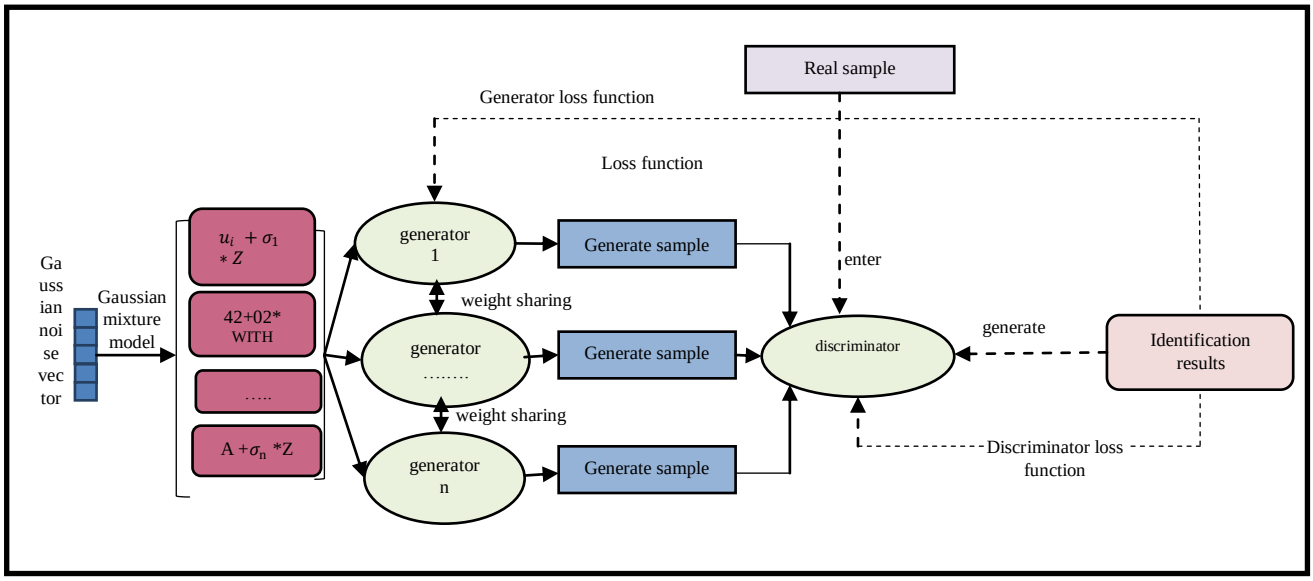

Figure 3: DMGAN model

At the same time, parameters for model training, such as learning rate, gradient clipping weight, and number of training iterations, were set. The learning rate was set to 0.001, and the gradient clipping range was set between [−0.01,0.01]. The model was trained for 500 iterations. The value of μ_i in the Gaussian mixture model was randomly selected from a uniform distribution in the range (−1, 1), while σ_i was fixed at 0.2 to avoid mode collapse (which could occur if σ_i were set to 0). Data generated by the generator was fed into the discriminator alongside the initial data [35]. When the discriminator's ability to distinguish between the two reached a probability of about 1/2, the model training was considered stable, and the training process was completed.

Throughout the 500 training cycles, the generator model was saved every 100 iterations. This approach not only captured the final trained model but also maintained diversity in the data, generating test cases with varying degrees of similarity and improving the diversity of generated data.

After completing the training process, the generator was tested to verify if it could produce data frames highly like real-world data. A reserved, real sample dataset was extracted for model validation. The discriminator was fed both actual and simulated data to determine whether the discriminator had reached the Nash equilibrium.

3.4 Mutation Strategy Based on Packet Operator Library

The mutation strategy is primarily used to determine where mutations will occur, while the mutation method decides the content of the mutation. Both works together on the packet sequence to achieve better mutation results, ensuring the generation of highly diverse and widely accepted test cases. This paper proposes a guided mutation operator library for packet sequences, based on known characteristics of existing vulnerabilities, to guide the mutation strategy and methods, allowing for efficient, rapid generation of valid cases within a controlled range, thus improving the efficiency of fuzz testing.

An analysis of the vulnerability library shows that, in digital twin within industrial control scenarios, abnormal communication packets can lead to issues like buffer overflow, format string vulnerabilities, integer overflow, logical errors, and denial of service. These vulnerabilities often result from improper handling of key fields in packets, unchecked boundary values, excessive length, or the presence of special characters. For example, buffer overflow occurs

when boundary checks for incoming data frames are absent, null pointer vulnerabilities arise from improper handling of null values, and out-of-bounds access happens due to the lack of boundary checks on data frame content.

In this study, mutation methods were developed based on the abnormal data frame collections and characteristics that triggered these vulnerabilities. The mutation methods were designed by altering the length characteristics of packets and introducing special characters in the packet content. For the mutation of characteristic fields in packets, we focused on boundary value mutations and similarity matching between fields. The initial mutation operators, defined according to known protocol field characteristics, are listed in Table 1.

Table 1: Mutation operator library

Protocol field	Mutation operator
TransactionID	Special symbols, special ASCII codes
Length	Greater than the exact length, less than the exact length, illegal length, special boundary value
UnitID	Legal but undefined identifier, illegal identifier, boundary value
FC	Illegal function code, legal but undefined function code by the device, random characters
Data	Extra-long string, single character, null value, illegal read value, illegal data address, random characters, space, separator (! # $ &?,..), formatted string (%d, %n, %x, %f,…)
Other types	Special ASCII code, directory traversal character

Considering the vulnerability triggering mechanisms and protocol field characteristics, a mutation strategy and method based on the mutation operator library and packet rules is proposed. A feature library of packets that are known to trigger vulnerabilities is established, and a mutation strategy is designed to perform different mutation operations on various regions of the packet, allowing for the quick identification of valid test cases.

3.5 Result Analysis and Feedback Optimization

The test cases generated by the model and those created through mutation are input into the test environment for evaluation. To ensure both a high reception rate and diversity in the data cases, a structured analysis of the protocol is performed. Fuzz testing scenarios are set up to align with the characteristics of digital twin within industrial control communication, and the test data frames are injected into devices or simulation programs for testing. The testing program or device interface is connected, and test data frames are progressively injected, with the operation of the test program or device being recorded, while abnormal communication frames are flagged.

To deepen the fuzz testing process, this study analyses and assesses the characteristics of the vulnerability-triggering packets and adjusts the mutation strategy and methods accordingly based on these characteristics. Additionally, abnormal response packets are analysed. If the packets causing a particular type of abnormal response are relatively uniform in the fuzz testing results, the mutation method is dynamically changed to improve the testing effectiveness. Once the specific field position causing the abnormal response is identified, mutations at that position are constrained to ensure the mutation result can trigger the target abnormal response. At the same time, mutations are still applied to other areas of the packet to potentially uncover other abnormal conditions.

In summary, by analysing and assessing the characteristics of abnormal packets, and dynamically adjusting mutation strategies and methods based on feedback from response information; while constraining specific field positions and applying mutations to other positions, the efficiency and accuracy of fuzz testing can be improved. This approach effectively identifies vulnerabilities.

To further enhance the diversity of test cases, a mixed iterative approach combining mutation and model generation is employed to filter out valid cases. This method enhances the mutation effect and helps evaluate the characteristics of vulnerability-triggering packets. The optimization process is shown in Figure 4.

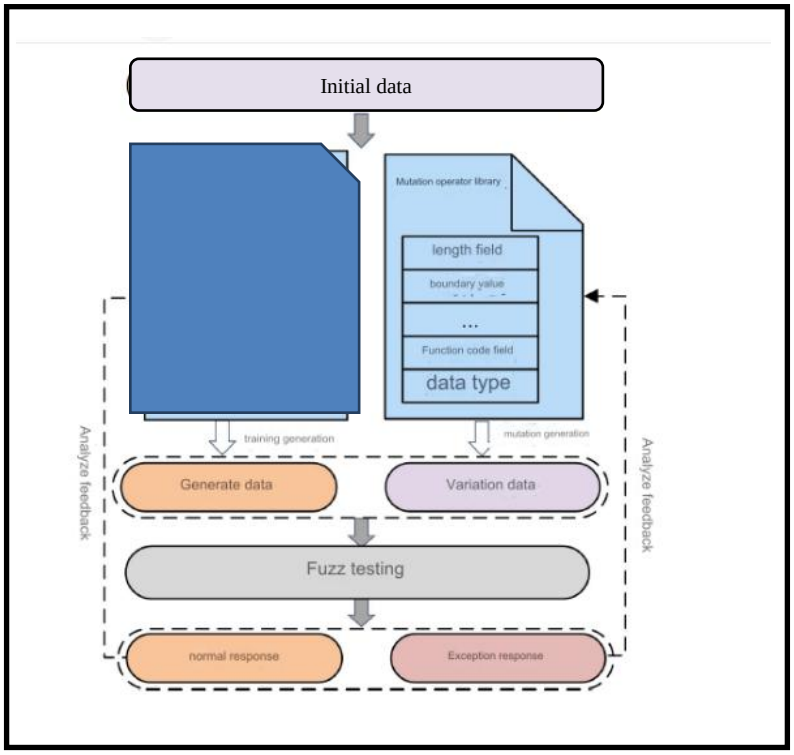

Figure 4: Feedback tuning process

4. EXPERIMENT

4.1 Experimental Setup

To validate the effectiveness of the proposed method in various aspects, we set up an experimental environment and performed fuzz testing to highlight the technical innovations [36]. The experiment mainly simulated and tested the communication processes of the commonly used industrial control network protocols, Modbus TCP [37] and S7 [38], to verify the effectiveness of the tests.

For Modbus TCP protocol fuzz testing, communication simulation tools adhering to Modbus protocol specifications were employed, including ModbusPoll v6.0.2, ModbusSlave v6.0.2, ModbusRSSim v8.20, and the serial port simulation tool Configure Virtual Serial Port Driver (VSPD). Siemens' S7-PLCSIM Advanced V3.0 high-function simulation software was installed to simulate the communication process of the S7 protocol by configuring the PG/PC interface and PLC IP addresses. Finally, the Wire shark tool was used to monitor abnormal communication between the master and slave stations.

The experiment used the industrial IoT control dataset provided by Lemay [40] for training. This dataset includes complete packet captures and files with malicious traffic labels, providing detailed information on the dataset generation. Additionally, the S7 protocol dataset was captured from the simulated environment for training. For model training, the DMGAN model was placed on a machine with 16 processors (Intel(R) Core(TM) i9-12900K CPU @ 3.20GHz), 32 GB of random-access memory, a 24-gigabyte Nvidia GeForce GTX 3090 Ti graphics card, and a 64-bit version of Windows 10 Professional. The results were compared with the training effects of models based on WGAN and WGAN-GP. The fuzz testing cases generated by the DMGAN model showed better performance across various dimensions.

4.2 Experimental Evaluation

To demonstrate the advantages of the proposed method, we introduced objective evaluation metrics such as test case reception rate, vulnerability detection capability, and generated data diversity to assess the overall effectiveness of the method. Additionally, the ability to generate test data and an evaluation of the benefits of the enhanced adversarial network model was conducted using the diverse data that was generated.

Data sent in an improper format will be rejected by the receiver, whereas data sent in an accurate format will be accepted. The test reception rate is a measure of how well the test cases that were generated were accepted by the test target. Here is its definition:

$$TA = \frac{n_a}{n_s} \times 100\% \quad (8)$$

The formula defines the test reception rate as follows:

- **ns**: the sum of all test cases that were transmitted
- **na**: the sum of all test cases that were received

During model training and mutation processes, adjusting model training parameters and mutation strategies can yield a higher test reception rate.

In vulnerability detection, the overall goal of the experiment is to discover more vulnerabilities using fewer test cases. The vulnerability detection rate reflects the ability to uncover vulnerabilities, making this metric the most direct standard for evaluating the effectiveness of the method. It is defined as follows:

$$AD = \frac{n_b}{n_c} \times 100\% \quad (9)$$

In this context, n_b stand for valid test cases, while n_c are denotes the overall quantity of test cases. To reflect the variability of the indicator over different phases, we count the number of erroneous data packets discovered through informal comparisons in every 100 test cases. This metric serves as the strongest indicator for evaluating the vulnerability correlation of the target program or device.

We also introduce the metric for the number of test cases generated per hour, which indicates how many run the model's tests produce in one hour. The formula is as follows:

$$\text{TCGPH} = \frac{n_{gc}}{\text{Hours}} \times 100\% \quad (10)$$

In this context, n_{gc} represents the number of generated test cases, and **Hours** gives the amount of time required to create these instances. To reflect the testing scope and capabilities of the proposed method, the diversity of generated data is considered an important metric. This measure highlights the model's diversity capability by concentrating on the number of categories within the produced data. If there are less data types in the produced set compared to the training set, it can be inferred that the model's performance is poor and needs tuning. Therefore, this metric serves as a training criterion for the model and can also be used as a selection standard for test cases.

4.3 Experimental Results

Table 2 lists the comparative results of this method against others.

Table 2: Experimental results

Method	Number of use cases	Test target	Test acceptance rate/%	Vulnerability detection rate/%	Number of vulnerability triggers
DMGAN	26000	Modbus Slave v4.3.4	89.4	6	298
		Modbus RSSim v8.20		3.02	187
WGAN	26000	Modbus Slave v4.3.4	74.3	4	97
		Modbus RSSim v8.20		2.61	63
WGAN-GP	26000	Modbus Slave v4.3.4	88.2	5.57	111
		Modbus RSSim v8.20		4.13	46
Peach mutation	26000	Modbus Slave v4.3.4	48.1	-	18
		Modbus RSSim v8.20			23
DMGAN combined mutation	26000	Modbus Slave v4.3.4	88.5	6.87	329
		Modbus RSSim v8.20		5.92	264

In this experiment, three models are compared with the proposed model: a model based on WGAN, a model based on WGAN-GP, and a model based on DMGAN. Additionally, the mutation method using the Peach tool is compared. A total of 26,000 test data packets generated by various methods are sent to the identically configured simulated slave,

Mod-busSlave, to observe their communication effectiveness. Compared to the models based on WGAN and WGAN-GP, the model based on DMGAN has a higher test reception rate, can trigger more anomalies, and generates cases with greater diversity. Traditional mutation methods are based on the mutation of known fields, and both their mutation methods and fields are random, resulting in lower test reception rates, and their vulnerability detection rates are not used as reference indicators.

Figure 5 shows the TA results during the model training period. As the training time increases, the TA rises, indicating that more generated data has the correct message format. In the stable phase, the test reception rate of the model based on DMGAN can reach 89.4%, which indicates higher format accuracy compared to the models based on WGAN and WGAN-GP. Although the model has been continuously adjusted, some data formats still remain incorrect. Initially, the test reception rate significantly increased; with ongoing iterations of training, the test reception rate slowly increased and eventually stabilized.

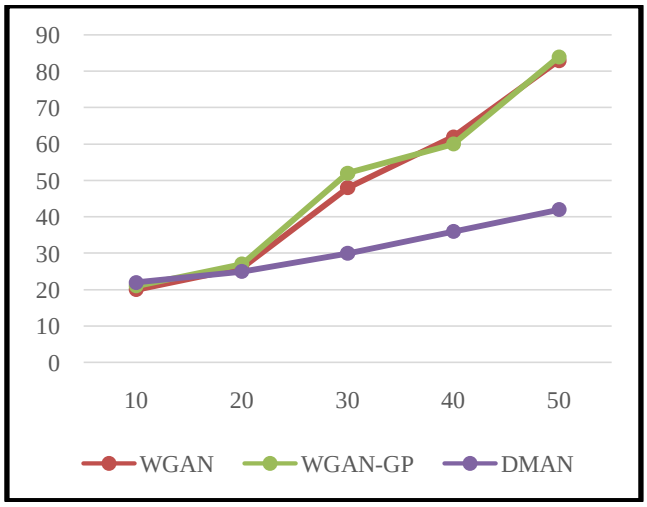

Figure 5: Test acceptance rate

Figure 6 shows the vulnerability detection capabilities of each model during the model training period. It can be observed that the proposed method shows an improvement in detection rates as training time increases, and the number of anomalous communications also rises, ultimately reaching a stable plateau. The level achieved by the proposed method in the experiment is not only related to the experimental approach but also to the testing targets. We used a Mod-bus Slave as our experimental object, and the improved DMGAN model demonstrated a stronger ability to detect errors compared to the models based on WGAN and WGAN-GP, validating the effectiveness and potential of the proposed method.

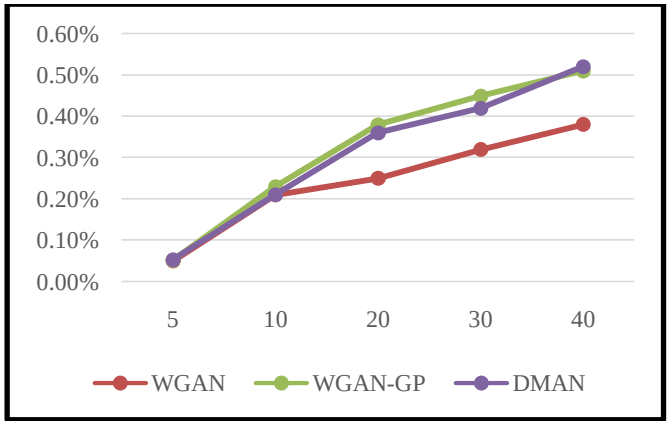

Figure 6: Vulnerability detection rate

In terms of case generation time, we conducted experiments to sequentially validate the efficiency of the DMGAN, WGAN, and WGAN-GP models. We found that when generating the same number of cases, the DMGAN model had a shorter generation time, as shown in Figure 7. In these experiments, the number of epochs was set to 30,000, and the generated datasets compared were the same, with the number of generated cases set to 1,000, 5,000, and 10,000.

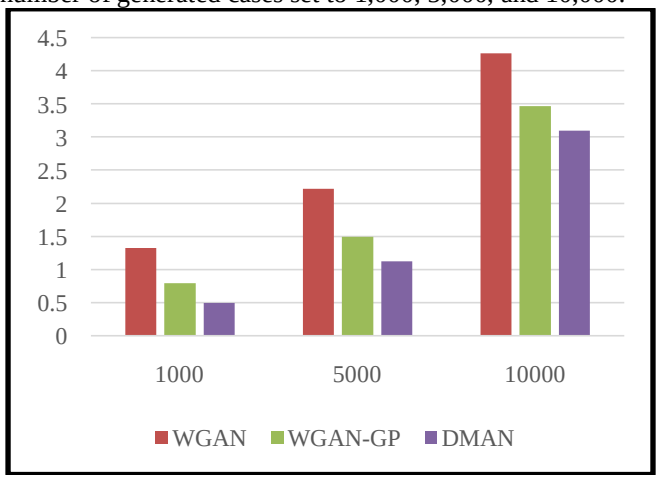

Figure 7: Example generation time

Upon examination and comparison, it was found that the original training data types retained their variety following training with both the WGAN and DMGAN models. Distinct variations between classes were visible in the DMGAN model's output. Data variety is thus better maintained by the DMGAN-based model than by the WGAN-based model. Abnormality detection capabilities tend to improve as data

type richness increases. The outcome is that the DMGAN-based model has a higher error detection rate. Figure 8 displays the K-Means clustering effect on the produced dataset.

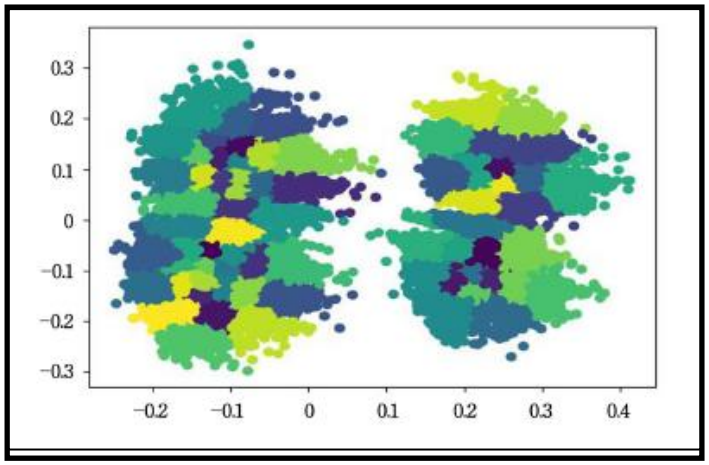

Figure 8: Clustering Effect of Generated Data

Table 3 lists some of the anomalies that occurred during communication testing. The first column of the table describes the protocol anomalies triggered by our method. It can be seen from the table that the fuzz testing method proposed in this paper not only ensures the ability to discover vulnerabilities but also increases the frequency of vulnerability detection, thereby enhancing testing efficiency.

Table 3: Anomalous Results

Exceptions	WGAN	WGAN-GP	DMGAN
Slave Crash	16	37	179
Station ID XX Off-Line	57	53	49
Software Caused Connection abort	23	19	23
Integer overflow	71	81	81
File not found	26	30	30
Illegal data address	119	135	169
Invalid Initialization	21	36	26
Illegal Function Code	160	197	297
Writer/ReadError	101	159	206

Below are detailed descriptions of some of the errors. When the test case attacks Modbus_Rssim, it causes a crash. After sending approximately 1,400 data frames, a prompt box indicating a program crash appears. Upon retesting these data frames by sending them to ModbusSlave, no anomalies occur. This indicates that there are defects in the implementation of Modbus_Rssim. Notably, when testing ModbusSlave, errors such as "writeError" and "ReadError" still allowed the execution of corresponding correct command operations. This is due to the read/write operations of the simulation program lacking functionality, but it demonstrates the ability of the proposed method to reveal software errors.

In further vulnerability mining processes, Modbus_Rssim prompts an abnormal message, displaying "Station number XX offline, no response sent." The testing software occasionally disconnects, and analysis shows that this is caused by a memory overflow leading to software crashes, indicating that the design of the simulator did not adequately consider data boundary filling situations. In further testing of the simulation environment, anomalies such as "function code abnormal," "data length mismatch," "integer overflow," and "address abnormal" were also discovered. Since these anomalies are common and have been explained in previous studies, only the test cases triggering these anomalies were recorded and provided for retraining the model to determine the message formats that caused the anomalies. Analysis of the anomalies revealed that the same anomalous behavior might be caused by different reasons; however, different

anomalous behaviours could also stem from the same cause. In the simulation experiments, due to the lack of source code for the testing targets, it was not possible to further determine the specific principles behind the anomalies.

Finally, this model was applied to Siemens S7 industrial control vulnerability mining. Unfortunately, Siemens S7 only performs packet sending and receiving without detecting anomalies in the communication process of the model. However, the ability to send and receive packets and monitor during the experiment demonstrates the feasibility of the method.

In summary, the test cases generated by the improved model outperform those of the previous model in both effectiveness and the ability to detect real vulnerabilities. With the increase in training iterations, the proportion of effective test cases generated by the adversarial model increases, and the protocol data frames that can trigger vulnerabilities also rise, confirming the feasibility of the method. Overall, this method can achieve vulnerability mining for unknown protocols with relatively ideal results, although it requires a long training time for complex industrial control protocols.

5. CONCLUSION

This paper employs proposed a novel model for generative adversarial networks (DMGAN) that uses mutation methods in conjunction with network protocol vulnerability mining. This model is combined with offline rapid mutation strategies applied to the vulnerability mining of digital twin within industrial control network protocols, enabling intelligent analysis and learning of the formats of vulnerability data messages in network communication processes without manual analysis, thereby achieving faster and more efficient discovery of network protocol vulnerabilities. Additionally, this method can adapt to the vulnerability mining process of unknown network protocols.

The improved model in this paper partially addresses the issue of mode collapse during the training process and is suitable for scenarios with smaller datasets. However, the model still has certain defects, such as long training times and dependency on the quality of the training dataset. The offline mutation strategy has not fully realized intelligent fuzz testing throughout the entire process and requires dynamic matching and adjustments. Future work could consider learning vulnerability rules, establishing complete memory and feature memory of these rules, and achieving more efficient and rapid intelligent vulnerability mining.

REFERENCES

[1] Paul Reedy,Interpol review of digital evidence for 2019–2022,Forensic Science International: Synergy,Volume 6,2023,100313,ISSN 2589-871X,https://doi.org/10.1016/j.fsisyn.2022.100313.

[2] SubirHalder, Thomas Newe,Radio fingerprinting for anomaly detection using federated learning in LoRa-enabled Industrial Internet of Things,Future Generation Computer Systems,Volume 143,2023,Pages 322-336,ISSN 0167-739X,https://doi.org/10.1016/j.future.2023.01.021.

[3] Shaashwat Agrawal, Sagnik Sarkar, OnsAouedi, GokulYenduri, KandarajPiamrat, MamounAlazab, Sweta Bhattacharya, Praveen Kumar Reddy Maddikunta, Thippa Reddy Gadekallu,Federated Learning for intrusion detection system: Concepts, challenges and future directions,ComputerCommunications,Volume 195,2022,Pages 346-361,ISSN 0140-3664,https://doi.org/10.1016/j.comcom.2022.09.012.

[4] DalilaRessi, Riccardo Romanello, Carla Piazza, Sabina Rossi,AI-enhanced blockchain technology: A review of advancements and opportunities,Journal of Network and Computer Applications,Volume 225,2024,103858,ISSN 1084-8045,https://doi.org/10.1016/j.jnca.2024.103858.

[5] AzimAkhtarshenas, Mohammad Ali Vahedifar, NavidAyoobi, BehrouzMaham, Tohid Alizadeh, Sina Ebrahimi, David López-Pérez,

[6] Federated learning: A cutting-edge survey of the latest advancements and applications,ComputerCommunications,Volume 228,2024,107964,ISSN 0140-3664,https://doi.org/10.1016/j.comcom.2024.107964.

[7] Mirna El Rajab, Li Yang, AbdallahShami,Zero-touch networks: Towards next-generation network automation,ComputerNetworks,Volume 243,2024,110294,ISSN 1389-1286,https://doi.org/10.1016/j.comnet.2024.110294.

[8] Mehdi Hazratifard, Vibhav Agrawal, Fayez Gebali, HaythamElmiligi, Mohammad Mamun,Chapter 12 - Review of using machine learning in secure IoThealthcare,Editor(s): Patricia Ordóñez de Pablos, Xi Zhang,In Information Technologies in Healthcare Industry,Accelerating Strategic Changes for Digital Transformation in the Healthcare Industry,AcademicPress,Volume 2,2023,Pages 237-269,ISBN 9780443152993,https://doi.org/10.1016/B978-0-443-15299-3.00007-5.

[9] Hang Thanh Bui, HamedAboutorab, ArashMahboubi, YansongGao, NazatulHaque Sultan, Aufeef Chauhan, Mohammad ZavidParvez, Michael Bewong, Rafiqul Islam, Zahid Islam, Seyit A. Camtepe, Praveen Gauravaram, Dineshkumar Singh, M. Ali Babar, ShihaoYan,Agriculture 4.0 and beyond: Evaluating cyber threat intelligence sources and techniques in smart farming ecosystems,Computers&Security,Volume 140,2024, 103754,ISSN 0167-4048,https://doi.org/10.1016/j.cose.2024.103754.

[10] Mohammad Kamrul Hasan, Rabiu Aliyu Abdulkadir, Shayla Islam, Thippa Reddy Gadekallu, NurhizamSafie,A review on machine learning techniques for secured cyber-physical systems in smart grid networks,EnergyReports,Volume 11,2024,Pages 1268-1290,ISSN 2352-4847,https://doi.org/10.1016/j.egyr.2023.12.040.

[11] Pengyong Li, Jiaqi Xia, Qian Wang, Yujie Zhang, MengWu,Secure architecture for Industrial Edge of Things(IEoT): A hierarchical perspective,ComputerNetworks,Volume 251,2024,110641,ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2024.110641.

[12] Sobhy M. Abdelkader, Sammy Kinga, Emmanuel Ebinyu, Jeremiah Amissah, Geofrey Mugerwa, Ibrahim B.M. Taha, Diaa-Eldin A. Mansour,Advancements in data-driven voltage control in active distribution networks: A Comprehensive review,Results in

Engineering,Volume 23,2024,102741,ISSN 2590-1230,https://doi.org/10.1016/j.rineng.2024.102741.

[13] Iqbal H. Sarker, Helge Janicke, Mohamed Amine Ferrag, AlsharifAbuadbba,Multi-aspect rule-based AI: Methods, taxonomy, challenges and directions towards automation, intelligence and transparent cybersecuritymodeling for critical infrastructures,Internet of Things,Volume 25,2024,101110,ISSN 2542-6605,https://doi.org/10.1016/j.iot.2024.101110.

[14] KhushiJatinkumarRaval, Nilesh Kumar Jadav, Tejal Rathod, Sudeep Tanwar, VrinceVimal, NagendarYamsani,A survey on safeguarding critical infrastructures: Attacks, AI security, and future directions,International Journal of Critical Infrastructure Protection,Volume 44,2024,100647,ISSN 1874-5482,https://doi.org/10.1016/j.ijcip.2023.100647.

[15] Mustapha El Alaoui, Khalid EL Amraoui, LhoussaineMasmoudi, Aziz Ettouhami, Mustapha Rouchdi,Unleashing the potential of IoT, Artificial Intelligence, and UAVs in contemporary agriculture: A comprehensive review,Journal of Terramechanics,Volume 115,2024,100986,ISSN 0022-4898,https://doi.org/10.1016/j.jterra.2024.100986.

[16] Umesh Kumar Lilhore, SurjeetDalal, SaritaSimaiya,A cognitive security framework for detecting intrusions in IoT and 5G utilizing deep learning,Computers&Security,Volume 136,2024,103560,ISSN 0167-4048,https://doi.org/10.1016/j.cose.2023.103560.

[17] Rambod Abiri, Nastaran Rizan, Siva K. Balasundram, ArashBayatShahbazi, Hazandy Abdul-Hamid,Application of digital technologies for ensuring agricultural productivity,Heliyon,Volume 9, Issue 12,2023,e22601,ISSN 2405-8440,https://doi.org/10.1016/j.heliyon.2023.e22601.

[18] MuaanurRehman, HayretdinBahşi,Process-aware security monitoring in industrial control systems: A systematic review and future directions,International Journal of Critical Infrastructure Protection,Volume 47,2024,100719,ISSN 1874-5482,https://doi.org/10.1016/j.ijcip.2024.100719.

[19] NtezirizaNkerabahiziJosbert, Min Wei, Ping Wang, Ahsan Rafiq,A look into smart factory for Industrial IoT driven by SDN technology: A comprehensive survey of taxonomy, architectures, issues and future research orientations,Journal of King Saud University - Computer and Information Sciences,Volume 36, Issue 5,2024,102069,ISSN 1319-1578,https://doi.org/10.1016/j.jksuci.2024.102069.

[20] Mingcan Cen, Xizhen Deng, Frank Jiang, Robin Doss,Zero-Ran Sniff: A zero-day ransomware early detection method based on zero-shot learning,Computers&Security,Volume 142,2024,103849, ISSN 0167-4048,https://doi.org/10.1016/j.cose.2024.103849.

[21] ElhamFazel, Mahmoud ZahedianNezhad, JavadRezazadeh, MarjanMoradi, John Ayoade,IoT convergence with machine learning &blockchain: A review,Internet of Things,Volume 26,2024, 101187,ISSN 2542-6605,https://doi.org/10.1016/j.iot.2024.101187.

[22] Deepak Adhikari, Wei Jiang, Jinyu Zhan, Danda B. Rawat, AsmitaBhattarai,Recent advances in anomaly detection in Internet of Things: Status, challenges, and perspectives,Computer Science Review,Volume 54,2024,100665,ISSN 1574-0137,https://doi.org/10.1016/j.cosrev.2024.100665.

[23] Yuxi Li, ShuxuanXie, Zhibo Wan, HaibinLv, Houbing Song, ZhihanLv,Graph-powered learning methods in the Internet of Things: A survey,Machine Learning with Applications,Volume 11,2023,100441,ISSN 2666-8270,https://doi.org/10.1016/j.mlwa.2022.100441.

[24] Mansi Gupta, Mohit Kumar, RenuDhir,Unleashing the prospective of blockchain-federated learning fusion for IoT security: A comprehensive review,Computer Science Review,Volume 54,2024,100685,ISSN 1574-0137, https://doi.org/10.1016/j.cosrev.2024.100685.

[25] Fatimah Aloraini, Amir Javed, Omer Rana, Pete Burnap,Adversarial machine learning in IoT from an insider point of view,Journal of Information Security and Applications,Volume 70,2022,103341,ISSN 2214-2126,https://doi.org/10.1016/j.jisa.2022.103341.

[26] Saqib Ali, Qianmu Li, Abdullah Yousafzai,Blockchain and federated learning-based intrusion detection approaches for edge-enabled industrial IoT networks: a survey,Ad Hoc Networks,Volume 152,2024,103320,ISSN 1570-8705, https://doi.org/10.1016/j.adhoc.2023.103320.

[27] VladyslavBranytskyi, MariiaGolovianko, Diana Malyk, VaganTerziyan,Generative adversarial networks with bio-inspired primary visual cortex for Industry 4.0,Procedia Computer Science,Volume 200,2022,Pages 418-427,ISSN 1877-0509,https://doi.org/10.1016/j.procs.2022.01.240.

[28] Feng Wang, YongjieGai, HaitaoZhang,Blockchain user digital identity big data and information security process protection based on network trust,Journal of King Saud University - Computer and Information Sciences,Volume 36, Issue 4,2024,102031,ISSN 1319-1578,https://doi.org/10.1016/j.jksuci.2024.102031.

[29] A. Mazumder, M.F. Sahed, Z. Tasneem, P. Das, F.R. Badal, M.F. Ali, M.H. Ahamed, S.H. Abhi, S.K. Sarker, S.K. Das, M.M. Hasan, M.M. Islam, M.R. Islam,Towards next generation digital twin in robotics: Trends, scopes, challenges, and future,Heliyon,Volume 9, Issue 2, 2023,e13359,ISSN 2405-8440,https://doi.org/10.1016/j.heliyon.2023.e13359.

[30] Manmeet Singh, DevNiyogi,Chapter 14 - Leveraging ML approaches for scaling climate data in an atmospheric urban digital twin framework,Editor(s): Saurabh Prasad, Jocelyn Chanussot, Jun Li,Advances in Machine Learning and Image Analysis for GeoAI,Elsevier,2024,Pages 315-346, ISBN 9780443190773,https://doi.org/10.1016/B978-0-44-319077-3.00019-5.

[31] Tarek Gaber, Joseph BamideleAwotunde, Mohamed Torky, Sunday A. Ajagbe, Mohammad Hammoudeh, Wei Li,Metaverse-IDS: Deep learning-based intrusion detection system for Metaverse-IoTnetworks,Internet of Things,Volume 24,2023,100977,ISSN 2542-6605,https://doi.org/10.1016/j.iot.2023.100977.

[32] Muhammad Hamza Zafar, Even FalkenbergLangås, FilippoSanfilippo,Exploring the synergies between collaborative robotics, digital twins, augmentation, and industry 5.0 for smart manufacturing: A state-of-the-art review,Robotics and Computer-Integrated Manufacturing,Volume 89,2024,102769,ISSN 0736-5845,https://doi.org/10.1016/j.rcim.2024.102769.

[33] Sabah Suhail, Mubashar Iqbal, Rasheed Hussain, Raja Jurdak,ENIGMA: An explainable digital twin security solution for cyber–physical systems,Computers in Industry,Volume 151,2023,103961,ISSN 0166-3615,https://doi.org/10.1016/j.compind.2023.103961.

[34] Van-Tam Hoang, Yared Abera Ergu, Van-Linh Nguyen, Rong-GueyChang,Security risks and countermeasures of adversarial attacks on AI-driven applications in 6G networks: A survey,Journal of Network and Computer Applications,Volume 232,2024,104031,ISSN 1084-8045,https://doi.org/10.1016/j.jnca.2024.104031.

[35] Xinzheng Feng, Jun Wu, Yulei Wu, Jianhua Li, Wu Yang,Blockchain and digital twin empowered trustworthy self-healing for edge-AI enabled industrial Internet of things,InformationSciences,Volume 642,2023,119169,ISSN 0020-0255,https://doi.org/10.1016/j.ins.2023.119169.

[36] Iqbal H. Sarker, Helge Janicke, Ahmad Mohsin, Asif Gill, LeandrosMaglaras,Explainable AI for cybersecurity automation,

intelligence and trustworthiness in digital twin: Methods, taxonomy, challenges and prospects,ICTExpress,Volume 10, Issue 4,2024,Pages 935-958,ISSN 2405-9595, https://doi.org/10.1016/j.icte.2024.05.007.

[37] Mehdi Saman Azari, StefaniaSantini, FaridEdrisi, Francesco Flammini,Self-adaptive fault diagnosis for unseen working conditions based on digital twins and domain generalization,Reliability Engineering & System Safety,2024,110560,ISSN 0951-8320,https://doi.org/10.1016/j.ress.2024.110560.

[38] Akram Hakiri, Aniruddha Gokhale, Sadok Ben Yahia, NedraMellouli,A comprehensive survey on digital twin for future networks and emerging Internet of Things industry,ComputerNetworks,Volume 244,2024,110350,ISSN 1389-1286,https://doi.org/10.1016/j.comnet.2024.110350.

[39] Samir Si-Mohammed, Anthony Bardou, Thomas Begin, Isabelle GuérinLassous, Pascale Vicat-Blanc,NS+NDT: Smart integration of Network Simulation in Network Digital Twin, application to IoTnetworks,Future Generation Computer Systems,Volume 157,2024,Pages 124-144,ISSN 0167-739X,https://doi.org/10.1016/j.future.2024.03.038.

[40] Muhammad Adil, Houbing Song, Muhammad Khurram Khan, Ahmed Farouk, Zhanpeng Jin,5G/6G-enabled metaverse technologies: Taxonomy, applications, and open security challenges with future research directions,Journal of Network and Computer Applications,Volume 223,2024,103828,ISSN 1084-8045,https://doi.org/10.1016/j.jnca.2024.103828.