Research on Real-Time Malware Variant Recognition and Adaptive Defense Mechanisms in Highly Dynamic Environments Driven by Generative Adversarial Networks


## Abstract

With the rapid development of information technology, the situation of network security is becoming increasingly severe, with the types and complexity of malware continuously increasing, posing significant challenges to the information security of individuals and organizations. To effectively protect users from attacks, it is essential to detect and respond to malware in a timely manner before it is executed. This study utilizes generative adversarial networks to find effective methods for malware sample generation and labeling, enriching the malware training dataset and improving the performance and integrity of malware detection models. Based on the dataset from generative adversarial networks, a real-time malware variant identification system is established in a high-dynamic environment to quickly and accurately detect continuously evolving malware. An adaptive defense mechanism is established that automatically adjusts security policies according to the characteristics of identified malware, thereby quickly and effectively protecting network security.



**Keywords:** Malware variant identification; Generative adversarial networks; High-dynamic environment; Adaptive defense mechanism

# Contents

# 1. Introduction

## 1.1 Research Background

Malware has been around in some form for as long as computer technology has been widely used. Malicious software refers to software or code that is installed and run on a user's computer or other terminal without explicitly prompting the user or without the user's permission, infringing the legitimate rights and interests of the user [1]. Common types of malware include viruses, ransomware, trojans, advertisements, etc. The identification and defense methods against malware are being updated, and the technology of malware is constantly iterating, becoming more and more complex and difficult to identify [2]. From PC to mobile, the types and functions of malware are diversified, and the functions are becoming more and more rich. In order to avoid being attacked by malicious software, the protection software usually detects and responds to new files. In a highly dynamic environment, the update speed of malicious software is greatly accelerated, and the protection software is confronted with new malicious code every time. In this case, it is more necessary to quickly identify and defend various new variants of malicious software [3]. The adaptive defense mechanism comes into being in this situation. The adaptive defense mechanism for malicious software can carry out health and identification of new malicious software, automatically adjust security policies according to the characteristics of malicious software in response to abnormal situations, and effectively combat the variants of malicious software [4]. Through machine learning technology, the flexibility and responsiveness of adaptive defense mechanisms can be significantly improved to actively respond to rapidly developing network threats [5].

The current malware identification methods are mainly divided into static and dynamic. Static malware detection does not require the execution of malware and can effectively identify existing malware types, but when encountering new types of malware variants, it is difficult to cope with rapidly updated malware types due to changes in software characteristics. Dynamic malware detection is identified by executing code in a simulated environment. The detection success rate is high, but it is

inefficient due to the large amount of simulation required and the time-consuming analysis [6]. It is necessary to identify malware variants in real time in a highly dynamic environment. Since the 21st century, the constantly updated malware variants have strong concealment and fast mutation rate, posing a serious threat to traditional detection techniques and the network environment of Internet users [7]. The recognition rate of former static identification technology is low in this case, and the feasibility is not good. The dynamic identification has a low efficiency, and the new malware cannot be tracked in a short period of time. This seriously affects the fluency of the users' internet environment. Therefore, a rapid and efficient method of identification is called for in this regard. In view of these requirements, the authors have proposed a dynamic-detection method with deep learning that analyzed the behavior pattern in real time and improved new malware detection capability to improve network environment security.

Generative Adversarial Network is a deep learning model that can be integrated into generating new malware samples, hence improving diversity for datasets, helping establish and optimize the identification defense systems. This paper seeks to find out the real-time identification of malware variants and an adaptive defense mechanism in a highly dynamic environment, which effectively detects the potential threat of malware, analyzes its behavior characteristics, and adjusts security policies effectively to deal with malicious attacks. This research is driven by adversarial network generation with the intent of identification and defense against real-time malware variants in highly dynamic environments. This research can provide new ideas in the identification of malware and contribute to the protection of network security.

### 1.2.Research Gap and Questions

#### 1.2.1Research Gap

(1) Limitations of past detection methods

In the past, the detection methods were mainly static and dynamic. Among them, the static detection method is difficult to deal with frequently updated malware variants and is not accurate enough. However, the recognition efficiency of dynamic detection in high dynamic environment is too low, which needs to be further

improved.

(2) Insufficient real-time monitoring and response ability to malware variables

The evolution of malware variants is very rapid and poses a great threat to network security. At present, how to implement effective monitoring of malware variables and respond to them quickly is an urgent problem to be solved.

(3) Lack of adaptive defense mechanism

After the malware is identified, the existing defense mechanisms are static or semi-static, and lack effective responses to the endless variants, so that the total number of malicious software cannot be quickly resolved, and a targeted defense mechanism is needed that can also change.

1.2.2 Research Questions

(1) How to effectively construct a high-quality malware sample set based on Generative Adversarial networks (GANs) to improve the effectiveness and recognition accuracy of model training?

(2) How to realize malware variation identification in a highly dynamic environment, and how to evaluate the performance of real-time malware detection system?

(3) Are the adaptive defense mechanisms of malware variants effective? Is its response speed timely?

1.3 Research Aims

This study uses generative adversarial networks to seek effective methods for malware sample generation and labeling, enrich the malware training data set, and improve the performance and completeness of the malware recognition model. Based on the data set of generative adversarial network, a real-time malware variant identification system is established in a highly dynamic environment to quickly and accurately identify the constantly updated malware. An adaptive defense mechanism is established to automatically adjust security policies based on the characteristics of malware after it is identified, so as to quickly and effectively protect the network security.

1.3Research Aims

This study uses generative adversarial networks to seek effective methods for malware sample generation and labeling, enrich the malware training data set, and improve the performance and completeness of the malware recognition model. Based on the data set of generative adversarial network, a real-time malware variant identification system is established in a highly dynamic environment to quickly and accurately identify the constantly updated malware. An adaptive defense mechanism is established to automatically adjust security policies based on the characteristics of malware after it is identified, so as to quickly and effectively protect the network security.

### 1.4.Significance of the Research

#### 1.Improving Malware Detection Efficiency

Through the use of Generative Adversarial Network (GAN) technology, rapid identification of malware variants can be achieved in high-dynamic environments, significantly enhancing the timeliness of detection. By utilizing GANs to generate high-quality adversarial samples, the training effectiveness of traditional classifiers is improved, thereby increasing the accuracy of identifying novel malware.

#### 2.Addressing Malware Diversity

Generative Adversarial Networks can produce diverse malware samples, aiding classifiers in learning richer features and enhancing their adaptability to variant malware. By automatically generating samples, the reliance on a large number of labeled samples is reduced, thus decreasing the time and cost associated with sample collection and labeling. Combining API execution sequence modeling and deep learning techniques, a novel dynamic detection scheme is proposed, promoting the advancement of malware detection technology. By employing deep convolutional neural networks (CNN) to extract high-dimensional features, the effectiveness of dynamic detection is enhanced, facilitating a more comprehensive analysis of malware behavior.

#### 3.Reinforcing Adaptive Defense Capabilities

The system can monitor and analyze process behavior in real time, automatically adjusting defense strategies to enhance the intelligence level of network security

protection. Upon detecting malicious behavior, suspicious processes are automatically isolated, effectively reducing potential damage and safeguarding user system security.

4.Advancing Research in Cybersecurity

This research integrates multiple fields, including machine learning, cybersecurity, and software engineering, promoting interdisciplinary convergence and development. The findings provide a theoretical foundation and practical reference for subsequent research on malware detection and defense mechanisms, driving the continuous evolution of technology.

## 2. Literature Review

2.12.1 Current Research Status of Malware Identification

Research on malware detection has received widespread attention both domestically and internationally. Researchers have proposed various detection techniques, including hybrid framework detection, behavioral modeling, code injection monitoring, and privilege control, to address the diverse range of malicious attacks. Static analysis methods identify potential malware by detecting key APIs and string features, while dynamic analysis simulates the execution behavior of malware in a sandbox environment to observe its actual performance.

Christensen et al. conducted static string analysis on imperative programming languages, specifically analyzing reflection code and dynamically generated SQL statements in Java programs. Using finite state automata (FSA) as the analytical model, they demonstrated the effectiveness of string analysis in capturing potential errors. Although computational linguistics methods have been referenced, they fall short in efficiency due to the lack of data source analysis and the need to determine finite states for each operation, making them unsuitable for vulnerability detection. Huang et al. minimized the number of inserted code using counterexample trajectories to improve the accuracy of error detection and reporting. They employed bounded model checking techniques, using variables that represent the legitimate information flow states to verify the program's security state. Various software testing techniques

were applied to web applications, combining user experience modeling with black-box testing and user behavior simulation; however, they failed to provide real-time protection for web applications and could not guarantee the detection of all defects.

In the area of dynamic feature analysis, Godefroid proposed detecting security vulnerabilities in web applications through runtime monitoring, using dynamic taint analysis methods to capture vulnerabilities in real time. However, this approach relies on real user input and may not cover all scenarios. Baylor implemented dynamic taint analysis to monitor the impact of user input on backend systems; although it provided greater accuracy, it incurred significant computational overhead and was affected by environmental changes. Wang et al. integrated static and dynamic analysis in an effort to improve the comprehensiveness of vulnerability detection. Despite the significant effectiveness of this hybrid approach, its implementation is complex and may lead to performance bottlenecks. Huang enhanced the security of web applications through user behavior analysis, identifying anomalous activities, but required a substantial amount of sample data for training and may not be sensitive enough to new types of attacks.

Zhao Binglin et al. processed malware API call graphs using convolutional neural networks (CNNs), constructing a receptive field for key nodes to reduce the complexity of the subgraph isomorphism problem, ultimately achieving a high accuracy of 93% in malware provenance analysis. However, the CNN structure used in this study is relatively simple, and the test samples were limited to 800 Windows 32-bit malware specimens. Mao Weixuan et al. introduced risk estimation into the malware data flow dependency network by analyzing the similarity and abnormality of process access behavior, proposing a malware detection method that achieves active learning by minimizing risk. This method successfully reduced the error rate to 5.55%, a decrease of 36.5% compared to traditional methods. However, the detection effectiveness of this approach is influenced by the choice of active learning step size, and it currently lacks the capability for adaptive step size adjustment. Overall, with the development of deep neural network technology, the research directions of

malware visualization and serialization have opened new avenues in the fields of static and dynamic analysis.

### 2.2 Current Research Status of Generative Adversarial Networks

Generative Adversarial Networks (GANs) have emerged as an important generative model, with significant applications in the fields of image and visual computing. Ledig et al. proposed the Super Resolution Generative Adversarial Network (SRGAN), which successfully generates photo-realistic super-resolution images with rich textures by utilizing VGG networks and residual networks. BEGAN demonstrated the capability of generating high-resolution facial samples from low-resolution photographs, encompassing a variety of poses and expressions. Santana et al. employed autoencoders and GAN-based costs to successfully generate images that resemble real driving scenarios, thereby supporting the development of autonomous driving technologies. Gou et al. explored the combination of synthetic and real images to enhance the accuracy of eye detection, although their research faced limitations regarding the synthetic and real images. Shrivastava et al. bridged the gap between synthetic and real images through SimGAN, refining the generated synthetic data. The Two-Path GAN (TP-GAN) proposed by Huang et al. can generate realistic frontal views from a single facial image, making it suitable for applications such as facial recognition. Zhu et al. improved CycleGAN by proposing an image translation method that does not require paired instances, applicable to various visual tasks. Overall, these studies illustrate the extensive application potential of Generative Adversarial Networks in image and visual computing, particularly in enhancing image quality, generating realistic scene images, and demonstrating effectiveness in specific tasks such as facial recognition and autonomous driving.

In recent years, Generative Adversarial Networks (GANs) have demonstrated broad application potential across multiple fields. The MalGAN algorithm proposed by Hu et al. utilizes the characteristics of GANs to effectively generate adversarial samples targeting malware detection models. This study analyzes the weaknesses of existing malware detection algorithms, particularly in black-box models where attackers can exploit subtle changes in input data to influence detection outcomes. By

constructing a generative network, MalGAN is capable of generating adversarial samples that are similar to real malware samples while minimizing the probability of detection. The results indicate that MalGAN successfully reduces the detection rates of various popular malware detection systems, and in some cases, renders detection systems completely ineffective.

The STGAN study by Chidambaram et al. delves into the generative process of style transfer, optimizing the style transfer task's generation effectiveness through the introduction of random network structures. This research primarily focuses on the environment of board games, utilizing style transfer techniques to create game board images with different styles, thereby assisting players in improving their strategic skills. The researchers designed a framework based on conditional GANs, allowing the generator to produce game board images of varying styles under specific conditions. Through extensive experiments, STGAN has demonstrated its potential in generating high-quality, diverse images, receiving positive feedback in both user experience and visual effects.

The medGAN developed by Choi et al. combines autoencoders and GANs for medical data generation, enabling the creation of synthetic data that closely resembles real Electronic Health Records (EHRs) while protecting patient privacy. This model excels in both the diversity and authenticity of data generation and competes effectively with real data in clinical predictive modeling tasks. Research indicates that the synthetic data generated by medGAN shows good efficacy across multiple disease prediction and diagnosis tasks, highlighting its significant potential in advancing medical research and promoting data sharing.

## 3. Research methods

### 3.1Literature Research

By collecting and organizing the research results of generating adversarial network and malware identification, this paper lays a foundation for the study of real-time malware variant recognition and adaptive defense mechanism in the low dynamic environment.

3.2 Theoretical Research

（1）Generative Adversarial Networks

Generative Adversarial Networks (GANs) can significantly enhance the detection capability of classifiers by fitting the distribution of real samples and generating new samples. The samples are divided into labeled and unlabeled parts, combining the advantages of supervised and unsupervised learning. The objective is to minimize the loss from both learning paradigms. A primary challenge faced by semi-supervised learning in practice is the scarcity of labeled data; in contrast to supervised learning, which relies on a large number of labeled samples to ensure effective training of the classifier and to avoid underfitting and overfitting issues, semi-supervised learning aims to make full use of the limited labeled data to improve model performance.

Utilizing unlabeled data can enhance the performance of the classifier, assuming that all samples are composed of data generated from a mixture of L Gaussian distributions:

$$f(x|\theta) = \sum_{l=1}^{L} \alpha_l f(x|\theta_l)$$

The mixture coefficients are defined by the summation formula $\sum_{l=1}^{L} \alpha_l = 1$, which serves to weight the contributions of different models. The samples are selected based on the conditional probability $(Pc_i \mid x_i, m_i)$. The maximum a posteriori (MAP) estimation yields the optimal classification:

$$h(x) = arg\max \sum_{j} P\,(c_i = k|m_i = j, x_i)P(m_i = j|x_i).$$

$$P(m_i = j|x_i) = \frac{\alpha_j f(x_i|\theta_j)}{\sum_{l=1}^{L} \alpha_l f(x_i|\theta_l)}$$

During the training process, samples are utilized for $P(ci = k|mj = j, xi)$and $P(mi = j|x)$, where the former relies on labeled data and the latter employs unlabeled data. It is assumed that the use of a large amount of unlabeled data can enhance the

estimation accuracy of the latter, thereby improving the generalization capability of the classifier. Unlabeled samples play a crucial role in semi-supervised learning. Under specific conditions, they can effectively simulate model parameters and improve the accuracy of model outcomes.

（2）Discriminative Networks

The discriminative network in Generative Adversarial Networks (GANs) is responsible for receiving two types of data: real data and data generated by the generative network. For the labeled data in the real dataset, the discriminative network not only needs to identify its authenticity but also to classify it correctly. In the case of unlabeled data, the discriminative network only needs to determine whether the data is real. For the fake data generated by the generative network, the discriminative network must effectively recognize and label it as fake data.

The Softmax function is widely used in machine learning. Given a>b, directly using the maximum function (max) would only select a, resulting in smaller values having almost no chance of being selected, leading to a phenomenon known as "the starvation of small values." To address this issue, the Softmax function calculates the selection probabilities based on the magnitude of the input values, ensuring that all inputs have a chance of being chosen. The Softmax value for the element Ei is:

$$Si = \frac{e^{Ei}}{\Sigma_j e^{Ej}}$$

The output of the discriminative network comprises three categories: 0 represents benign samples from the real data, 1 represents malicious samples from the real data, and 2 represents fake samples generated by the generative network. For unlabeled real data, the objective of the discriminative network is to maximize the selection of non-fake samples. Let pmodel(y∈{0} | x) denote the probability that a sample is a benign sample from the real data, pmodel(y∈{1}|x) denote the probability that a sample is a malicious sample from the real data, and pmodel(y∈{2}|x) denote the probability that a sample is a fake sample generated by the generative network. The loss function is:

$$Loss_D = L_{supervised} + L_{unsupervised}$$

$$L_{supervised} = -E_{z\sim p_{data}} log p_{model}(y|x, y \in \{0,1\})$$

$$\begin{aligned} L_{unsupervised} &= -E_{x\sim p_{data}} \log\left(1 - p_{model}(y|x, y = 2)\right) \\ &+E_{x\sim G(z), z\sim noise} \log p_{model}(y|x, y = 2) \end{aligned}$$

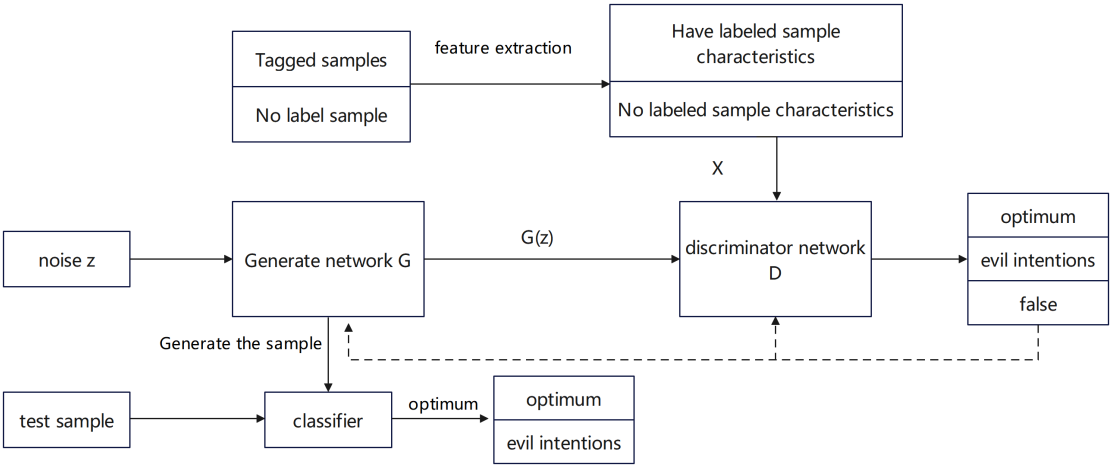
if $\mathrm{D}(x) = 1 - p_{model}(y|x, y = 2)$,

then $L_{unsupervised} = -E_{x\sim p_{data}} \log D(x) + E_{x\sim G(z), z\sim noise} \log\left(1 - D\left(G(z)\right)\right)$.

The fundamental function of the discriminative network is to determine the source of the input samples and estimate the probability that they are real or generated samples, typically outputting a binary classification result: true or false. In specific application scenarios, this discriminative network needs to be extended to three output classes, including benign samples, malicious samples, and fake samples. This extension allows the network to not only fulfill the role of a discriminator but also to incorporate the functionality of a classifier, enabling it to simultaneously handle both the authenticity and category of the samples.

3.3Experimental Research

To verify the effectiveness of the method proposed in this paper, the experimental section is designed with three main components: data crawling, GAN model training, and the implementation of a proxy tool. First, a large amount of malicious and benign malware code was collected using web crawling techniques, and the data underwent regular processing to ensure its quality and usability. Next, the concept of GAN was applied to sample generation, conducting five sets of experiments using classifiers such as RF (Random Forest), LR (Logistic Regression), DT (Decision Tree), SVM (Support Vector Machine), and KNN (K-Nearest Neighbors), observing the trend of recognition accuracy as the sample size varied to evaluate the method's effectiveness. Finally, based on the theoretical research, a server-side proxy tool was developed, integrating the trained model to validate the effectiveness of the adaptive defense mechanism in practical applications.




## 4. Timeline

| Task ID | Task Name | Sep-24 | Mar-25 | Sep-25 | Mar-26 | Sep-26 | Mar-27 |
|---|---|---|---|---|---|---|---|
| T01 | Literature study | | | | | | |
| T02 | Theoretical collection | | | | | | |
| T03 | Sample collection and preprocessing | | | | | | |
| T04 | Generative adversarial network (GAN) construction | | | | | | |
| T05 | Feature extraction and modeling | | | | | | |
| T06 | Thesis writing and submission | | | | | | |

# References

[1] Alhashmi A A , Darem A A , Alanazi S M ,et al.Hybrid Malware Variant Detection Model with Extreme Gradient Boosting and Artificial Neural Network Classifiers[J].计算机、材料和连续体(英文), 2023, 76(9):3483-3498.

[2] Kazi M A , Woodhead S , Gan D .Detecting Zeus Malware Network Traffic Using the Random Forest Algorithm with Both a Manual and Automated Feature Selection Process[J]. 2023.

[3] Zhu H , Wei H , Wang L ,et al.An effective end-to-end android malware detection method[J].Expert Systems with Applications, 2023, 218:119593.

[4] Xiang Y , Li D ,XinyiMeng,et al.ResNeSt-biGRU: An Intrusion DetectionModel Based on Internet of Things[J].Computers, Materials & Continua, 2024, 79(4):1005-1023.

[5] Vasilellis E , Botsos V , Anagnostopoulou A ,et al.Gaming the system: tetromino-based covert channel and its impact on mobile security[J].International Journal of Information Security, 2024, 23(4):3007-3027.

[6] Liu L , Wang Y , Liao S ,et al.CL-GCN: Malware Familial Similarity Calculation Based on GCN and Topic Model[J]. 2022.

[7] Li S , Tang Z , Li H ,et al.GMADV: An android malware variant generation and classification adversarial training framework[J].Journal of Information Security and Applications, 2024, 84.