Research Article

An Intrusion Detection Model Based on Improved ACGAN in Big Data Environment

Jianfeng Liao

Department of Computer, Wenhua University, Wuhan, Hubei 430074, China

Correspondence should be addressed to Jianfeng Liao; jazz9812@163.com

Received 22 February 2022; Revised 12 March 2022; Accepted 21 March 2022; Published 9 May 2022

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

With the development of big data technology, network intrusion problems against server vulnerabilities emerge one after another. To improve the accuracy of intrusion detection, this paper designs an intrusion detection platform based on the ACGAN (auxiliary classifier generative adversarial network) model in a big data environment. Firstly, by introducing a self-attention mechanism, the global characteristics of attack samples are extracted to improve the quality of generated samples. Then, by adding a gradient penalty, the model’s convergence speed and training stability are improved. Finally, this method enhances and expands the attack samples and verifies the dataset. The experimental results show that compared with other comparison methods, the overall detection accuracy of this system is higher, and the false-positive rate and false-negative rate are lower.

1. Introduction

With the rapid development of computer and network technology, people’s life depends more and more on the convenience brought by electronic equipment. Still the accompanying computer security problems are becoming more and more acute. According to the statistics of vulnerability data of Windows platform in recent years, the amount of vulnerability submission of Windows host system generally shows an upward trend year by year [1], and the intrusions launched against host vulnerabilities emerge one after another. How to effectively detect intrusion has become one of the focuses of network security research [2]. Generally speaking, according to the source of detection data, intrusion detection can be divided into network-based intrusion detection and host-based intrusion detection. Network-based intrusion detection uses the original IP packet as the data source to detect intrusion. Host-based intrusion detection generally finds intrusion by detecting systems, events, and system logs [3].

With the application of machine learning technology in various research fields, the intrusion detection model based on machine learning has gradually become the current research trend [4]. However, compared with deep learning models, traditional machine learning models such as Bayesian algorithms and decision trees have some deficiencies in data processing and feature association with unclear features or complex internal constraints [5]. Therefore, intrusion detection based on the deep learning model has become one of the research hotspots. Javaid et al. [6] proposed a self-learning technology based on deep learning, which learns good feature representation from unmarked data and then classifies intrusion. Yin et al. [7] used recurrent neural network for intrusion detection to improve the accuracy of detection. Qu et al. [8] proposed an intrusion detection model based on deep confidence network. Shone et al. [9] proposed an asymmetric deep autoencoder (NADE) based on unsupervised feature learning. It can be seen that the current intrusion detection system based on deep learning mainly focuses on the automation of high-dimensional data feature extraction, dimensionality reduction of high-dimensional data features, and improving the ability of sample recognition. Also, most studies use NSL-KDD as their training and testing datasets [10].

Although intrusion detection based on deep learning can effectively detect malware, malicious behavior, and malicious code, there are still the following limitations. (1)
In the training process, the attack samples are far less than the standard samples, resulting in the imbalance of the detection model and the inability to detect malicious attacks correctly. (2) With the development of malicious attack technology, the attack methods of attackers are constantly changing. Learning through the known intrusion knowledge base will make the model unable to detect the unknown attack data [11]. Therefore, researchers introduced generative adversarial networks (GANs) to generate usable attack data and enhance the training dataset to improve the performance of the detection model [12].

Jin et al. [13] proposed a self-evolving GAN model based on the game idea. At present, the model has been successfully and widely used in image classification and sample generation. It is mainly used to solve the problems of unstable training, pattern collapse, and sample generation. It has been studied to expand the samples of the malicious code base through the GAN to solve the problem of aging attacks caused by the evolution of intrusion means [12]. Some GAN-based detection models have also been proposed, such as t-GAN [14] for detecting malicious code, t-DCGAN [15] for improving the stability of t-GAN model training process, BOT GAN [16] for detecting botnets, and CF-GAN [17] for detecting online payment fraud.

Aiming at the problem of low diagnostic accuracy, this paper proposes a data enhancement method based on ACGAN. The two-dimensional convolution network model is adopted to effectively reduce the amount of network training parameters and improve the convergence speed of the network. At the same time, gradient punishment is used instead of weight clipping to overcome the problems of gradient disappearance and mode collapse and enhance the stability of model training. By introducing the attention mechanism into the generator and discriminator of ACGAN, the global features of the samples are fully extracted, and the generation quality and learning efficiency of the model are improved. Experiments show that this method can balance the impact of unstable data on classification accuracy and effectively reduce the misjudgment rate.

2. Related Work

2.1. Generating Countermeasure Network. GAN consists of opposing deep convolution networks, generator A, and discriminator D, respectively. A receives a set of noise satisfying the joint Gaussian distribution \( U \_k \) and \( K \) is its input vector. The mapping relationship between \( U \_k \) and actual data distribution \( U \_data \) is established through a multilayer neural network [18], and new samples as close to the actual data as possible are generated. Then, it is sent to D and the actual sample to distinguish whether the data is the real quantity or the generated quantity. G and D compete with each other, forming a dynamic process of the zero-sum game [19]. When both losses reach the minimum, Nash equilibrium is reached. (1) shows the objective function of GAN:

\[
\min \min Q(D,A) = E_{U\_data} [\log_2 D(i)] + E_{U\_data} [\log_2 (1 - D(A(k)))]
\]

where \( E[\cdot] \) is the expectation of the corresponding distribution, \( A[k] \) generates data for the generator, and \( D[\cdot] \) is the judgment result of the discriminator.

The loss functions \( L_A, L_D \) of generator and discriminator are as follows:

\[
L_A = E_{U\_data} [\log_2 (1 - D(A(k)))]
\]

\[
L_D = E_{U\_data} [\log_2 D(i)] + E_{U\_data} [\log_2 (1 - D(A(k)))]
\]

2.2. Auxiliary Classification Generation Countermeasure Network. The traditional GAN employs unsupervised learning, and the model is too accessible, resulting in an uncontrollable training process. CGAN adds the auxiliary classification label \( c \) to the generator and discriminator, uses \( c \) to guide the direction of data generation, and realizes supervised learning [20]. ACGAN is improved based on conditional generation countermeasure network. The label information only acts on the generator, and the auxiliary classifier (C) is introduced into the discriminator to distinguish the category of samples. Therefore, the loss function of ACGAN can be divided into loss function \( L_S \) representing the authenticity of data and loss function \( L_C \) representing the accuracy of data classification:

\[
\begin{align*}
L_S&= E_{U\_data} [\log_2 D(i)] + E_{U\_data} [\log_2 (1 - D(A(k)))]

L_C&= E_{U\_data} [\log_2 D(c)] + E_{U\_data} [\log_2 (1 - D(A(c)))]
\end{align*}
\]

2.3. Self-Attention Generation Confrontation Network. Most GANs are based on a convolutional neural network. Because the local acceptance domain of convolution calculation has a fault limit, it can only calculate the data features in a specific neighbourhood. Still, it cannot mine the feature relationship between long-distance spatial regions, so the calculation efficiency is low. SAGAN adds a self-attention module in G and D, which can realize the modelling of remote sample dependency. The generator can mine feature information from all locations to generate sample details. At the same time, the discriminator can judge whether the feature details of the distant parts in the proper and false samples are consistent. It can master more comprehensive sample information and improve the network’s overall performance. Figure 1 shows the feature calculation flow of self-attention module.

In Figure 1, \( \otimes \) represents matrix multiplication, and each row is activated using softmax. After the sample features from the previous hidden layer \( i \) are multiplied by different weight matrices, they are first converted into the sum of two feature spaces \( f(i) \) and \( b(i) \) to calculate the attention map, the correlation of two feature spaces.
3. Algorithm Model

3.1. Improved ACGAN Model. ACGAN can control the direction of sample generation in the generation process through an auxiliary classifier to generate high-quality results. However, due to the limited size of the convolution kernel, we can only learn the relationship between the local region of the sample. The learning efficiency of the model is low, and details may be lost. Based on the supervision idea of ACGAN, the self-attention mechanism is added to A and D to help the model capture the relationship between the long-distance features of the sample. Using Wasserstein distance measure to generate the difference between samples and real samples, an improved ACGAN model is constructed to generate high-quality attack samples for diagnosis. The weight clipping can meet the 1-Lipschitz condition. However, due to the restriction of weight, the ability of network learning decreases. Also, the weight clipping is easy to set improperly, which will lead to gradient explosion or disappearance. A gradient penalty term is established, as shown in (8), to replace the weight clipping to realize the 1-Lipschitz condition to solve the above problems. The gradient penalty is a soft constraint, which will control the gradient around 1. The controllability is strong, the model is stable, and the above gradient problem is alleviated. At the same time, L1 regularization is added to the generator to improve the generalization ability of the generated model and alleviate overfitting.

\[ L_{gu} = \lambda \varepsilon_i \sim u \left( \| \nabla D (\tilde{i}) \|_2 - 1 \right)^2, \]  

where \( \tilde{i} \) represents the interpolation sample between the real image and the generated image. Therefore, the loss function of the improved ACGAN framework is divided into three parts, that is, generator loss \( L_A \), discriminator loss \( L_D \), and classification loss \( L_C \), respectively.

\[ L_A = -E_{k-u_{data}} [D(A(k), c)], \]
\[ L_{gu} = E_{i-u_{data}} [D(i, c)] - E_{k-u_{data}} [D(A(k), c)] + \lambda \varepsilon_i \sim u \left( \| \nabla D (\tilde{i}) \|_2 - 1 \right)^2, \]
\[ L_C = E_{c-u_{data}} \left[ \log_2 D(c) \right] + E_{c-u_{data}} \left[ \log_2 (1 - D(A(c))) \right], \]

where the generator and discriminator must meet the 1-Lipschitz condition, that is, \( D_m L \leq 1 \), and \( A_m L \leq 1 \).
3.2. Improved ACGAN Structure. The improved ACGAN model framework is shown in Figure 2. The deconvolution layer adopts batch normalization and ReLU activation function, and the output deconvolution layer adopts tanh activation function. The 100-dimensional random noise \( k \) and 4-dimensional label \( c \) are connected to the input generator, and the dimension expansion is realized through 3-layer deconvolution and then input to the self-attention layer. The feature details are enriched by calculating the self-attention feature map, and then the samples are output after two layers of deconvolution.

The discriminator model adopts a structure symmetrical to the generator model. The output layer uses the softmax activation function to classify the data. To adapt to the GP gradient penalty term, the discriminator removes spectrum normalization and batch normalization. For each convolution layer, LeakyReLU is used as the activation function, and dropout is added to reduce the calculation parameters of the model and alleviate overfitting.

3.3. Diagnostic Training Steps. The ACGAN-based data enhancement framework proposed in this paper is used for attack sample diagnosis. The process is shown in Figure 3.

1. The attack samples are obtained from the experiment, preprocessed, and normalized to obtain the actual dataset. By establishing the mapping relationship between the noise sample distribution \( u_k(\mathbf{k}) \) and the actual sample distribution \( n_{\text{data}}(\mathbf{i}) \), the generator will generate a batch of sample data mixed with the actual attack samples and send them to the discriminator. The discriminator discriminates between true and false and backpropagates the gradient information to update the parameters of the network.

2. As a round of training process, the above steps update the generator once after updating the discriminator \( z \) times. The two update alternately until they reach Nash equilibrium.

3. Use the sample data generated in step (2) to expand the original unbalanced dataset, obtain the balanced sample dataset, and send it to the classifier for diagnosis.

4. Design of Intrusion Detection Platform

4.1. Establishing Database Intrusion Detection Model. Because the designed intrusion detection system pays attention to real-time intrusion monitoring, setting up a database intrusion detection model is necessary. The core of the model is a hybrid detection engine, which has the functions of anomaly detection and misuse detection. The audit data can be transferred to the intrusion detection platform for real-time audit and reanalysis. Judge whether the data are expected, abnormal, or attacked, respond to the operation results, and report to the administrator. The management personnel carries out the subsequent processing. The detection model based on this is shown in Figure 4.

As can be seen from Figure 4, the functions of the main components of the detection model are as follows. In the event generation function, the detection log needs to wait for the generation of detection records before data collection, but the detection records are created and stored in the detection log. Extracting detection records from the database can realize the parallelization of data detection and data acquisition. The anomaly detection unit and intrusion detection unit use the serial mode of detection to read the data from the data collector, respectively, and then detect in combination with the rules in each detection rule base. The behavior pattern rule database is the exception detection storage module, which can generate and gradually update the behavior pattern of normal users according to the data mining algorithm of association rules. The recorder can cache the action results generated by the intrusion detection unit and the anomaly detection unit. The alarm can take the abnormal behavior detected by the intrusion detection unit and the abnormal detection unit as the alarm signal and send it to the reflection platform. The alarm takes corresponding actions according to the abnormal behavior.

4.2. Setting Up Database Security Mechanism Based on Cloud Computing. Cloud computing can identify all kinds of information in the database and establish the security mechanism of the database. Firstly, the security mechanism of the intrusion detection system of the database includes a configuration layer, which can ensure that the data can be accessed only with appropriate authorization. Secondly, the database’s security principal can connect to the server and request access or control one or more database objects after passing the security account authentication. Because multiple databases created by different users are often stored on one server, even if a user passes the security account authentication, it is impossible to access all databases, which further limits the user’s access scope and operation type. Therefore, database security is realized through an authorization mechanism. Only the owner of the database can access the objects in the database, which also plays an important role in the security and stability of the database. Unauthenticated user connection requests need to be authenticated using Windows authentication.

4.3. Designing the Main Functions of the Database Intrusion Detection Platform. The anomaly detection unit is designed based on relevant rules. Using data mining algorithm, the user’s behavior can be preliminarily detected and stored in the behavior pattern rule base in the form of rules. In the learning stage, the influencing factors of rule generation include support and confidence. In the detection stage, the factors affecting rule generation also include generating abnormal data. Security experts create the rules of the rule base based on experience and detection strategy. Disk space management can be configured in the space management module. The space management module can display disk usage, space used value, and free space value. Set the number of records stored in the event library simultaneously. You can adjust the amount of data recorded in the event library
between the maximum upper limit and the maximum lower limit. The detection and maintenance module can query the status of historical events in the database’s intrusion detection time and anomaly detection results and count the times of each intrusion or anomaly in a specific unit time.

### 4.4. Realizing Database Intrusion Detection

Considering the future upgradeability and maintainability of the system and the portability, the system designed in this paper adopts graphical user interface (GUI). The interface of database intrusion detection system is divided into login interface and main interface. To create a graphical user interface, first create a framework and add various Swing components based on the framework. Before adding components, you need to get the ContentPane container of JFrame first and then add all components. The login interface requires the system administrator to enter the correct user name and login password to enter the main interface to ensure system security. The menu bar is mainly used for real-time monitoring, detection and maintenance, window configuration management, and attack information transmission. At present, the mining technology based on Boolean association rules can only be used in a transactional database. Therefore, numerical association mining technology is more widely used than Boolean association rules. When studying related association rules in a relational database, the database
can contain binary attributes and many classifications or numerical attributes. To realize data mining, we first convert these attributes into Boolean attributes and then map each subset to an entity. Users can use the system by entering the system console. If the display fails to match, it means that the user does not exist or the password is entered incorrectly, and the system console cannot be accessed.

5. Experiment and Analysis

To validate the intrusion detection model, build a simulated experimental environment and use the deep learning framework based on tensorflow-GPU1.13 Keras2.2.4 to simulate. The operating system is Windows10, Intel i5-6300HQ 4-core processor is used, the memory size is 8 GB, and NVIDIA GTX960 graphics card is used to speed up the running speed of GPU.

5.1. Data Digitization and Normalization. The dataset used in the experiment is the UNSW-NB15 dataset, which is the latest dataset in intrusion detection. The dataset was created by the Australian Cyber Security Centre (ACSC) in 2015, which covers a large number of low occupancy and deeply structured network traffic information. It represents the current network traffic mode and adjusts the training and test sets. It is more suitable for simulating the current complex network environment and getting better test results. The dataset has 9 different types of modern attacks and 49 features. There are 5 more attack types than NSL-KDD, including 2540044 samples. It contains 9 types of attacks: Fuzzers, DoS, Analysis, Reconnaissance, Exploit, Shellcode, Worm, Backdoor, and Generic. Each piece of data has 47-dimensional characteristics, one specific attack category identifier, one attack, and a normal category identifier.

(1) The unique heat coding is used for digitizing the character features.

To convert character-type features according to the data set protocol type, service, and status, it is necessary to convert character-type features into numerical features through one-hot encoding. Protocol type includes three categories of TCP, UDP, and ICMP, which are converted to numerical type. The features are three-dimensional features [1, 0, 0], [0, 1, 0], [0, 0, 1]. For service, there are 70 cases, so the values are converted to 70-dimensional features, and for the state, there are 11 cases. Therefore, the values are converted to 11 dimensions. After numerical processing, the whole dataset becomes 130-dimensional numerical features, of which the first 128 dimensions are features. The last two dimensions are class labels.

(2) Normalization Processing. After processing the character features, the features in the dataset are divided into continuous and discrete attributes, making the differences between different features extensive. Therefore, it is necessary to normalize the eigenvalues in the dataset to the \([-1, 1]\) interval. In this paper, the min-max standardization method is used for normalization, which only compresses the data. It does not change the original information of the data. The conversion formula is as follows, where \(j_{\text{min}}\) and \(j_{\text{max}}\) represent the maximum and minimum values of the original eigenvalues, respectively, and \(y\) represents the eigenvalues before conversion.

\[
j' = \frac{j - j_{\text{min}}}{j_{\text{max}} - j_{\text{min}}}.
\]

5.2. Evaluation Index. To verify the detection performance of different models, it is necessary to calculate according to the detection sample category and actual category of the model. This paper mainly uses accuracy (ACC), mean square error (MSE), false-positive rate (FPR), and false-negative rate (FNR) to evaluate the model. It is assumed that \(n_x\) represents the actual category of the \(x\)-th sample, \(n_x'\) represents the detection category of the \(x\)-th sample, and the sample set is \(t\). There are \(u\) missing events and \(f\) false events. When \(n_x = n_x'\), it means that the model detection is accurate. Then, the accuracy on the sample set \(t\) is

\[
\text{ACC} = \frac{1}{t} \sum_{x=1}^{t} L(n_x = n_x'),
\]

where \(L(i)\) represents the detection function. When all the detection result categories in the sample set are the same as the actual category of the sample, the accuracy is 1. Mean square error is a method to measure the average value of error and evaluate the change degree of data. The calculation formula is defined as

\[
\text{MSE} = \frac{1}{t} \sum_{x=1}^{t} (n_x - n_x')^2.
\]

The false-positive rate is defined as

\[
\text{FPR} = \frac{f}{t}.
\]

The underreporting rate is defined as

\[
\text{FNR} = \frac{u}{t}.
\]

Because of the complexity of network intrusion detection data, it is difficult to define the standard of evaluating the model. Therefore, this paper comprehensively compares the test results of each model through ACC, MSE, FPR, and FNR to verify the accuracy and stability of the model.

5.3. Results and Analysis. Using ACGAN to expand the attack samples of Analysis, Shellcode, Backdoor, and Worm in the training set, respectively, analyze the impact of the expansion proportion of different training sets on the detection rate of a few classes. When the proportion is 0%, 40%, 80%, and 120%, conduct four groups of comparative experiments. The results are shown in Table 1. The experiment shows that the accuracy rate is the highest when the
expansion ratio is 80%, so 80% is selected for this expansion ratio.

In order to prove the effect of the intrusion detection model, the same training set and test set are selected, and three different algorithms are used to compare with the algorithm in this paper. The comparison of model performance results is shown in Table 2.

In order to verify whether the algorithm performance is improved after ACGAN expands a few samples, it is compared with the algorithms in literature [21], literature [22], and literature [23]. As can be seen from Figure 5, the classification accuracy is improved by 1.39% after supplementing a few samples through the ACGAN model.

Figure 6 compares the mean square error of the algorithms in this paper, literature [21], literature [22], and literature [23]. As can be seen from Figure 6, the mean square error of the algorithm in this paper is the lowest. Compared with the literature [21], it is reduced by 87.36%. Compared with the literature [22], it is reduced by 75.21%. Compared with the literature [23], it is reduced by 39.59%. Therefore, the network intrusion detection model based on this algorithm greatly reduces the error of network intrusion detection.

The same training and test set are selected to verify the advantages of the network intrusion detection model proposed in this paper. The complexity of this model is compared with the CNN, LSTM model, and DBN model. The results are shown in Figure 7. It can be seen from the figure that with the increase of the number of test samples, the detection time of the model in this paper is less than that of LSTM and DBN models, and the time difference gradually increases. Compared with the CNN model, the detection time is slightly higher, but the overall difference is limited. Then, we compare the test results of this method with other machine learning methods, as shown in Table 3.

By comparing with the current commonly used intrusion detection algorithms, it can be seen that some classical machine learning algorithms, such as SVM and KNN, have poor experimental results and high overall false alarm rates. In contrast, the deep learning algorithm is better than SVM and KNN in overall indicators. Still, the detection rate for rare attacks is not high. Compared with other methods, the overall accuracy of the algorithm proposed in this paper is improved by 7.8%, the false-positive rate is reduced by 8.89%, and the false-negative rate is reduced by 12.05%. Also, for rare attacks such as Analysis, Shell code, Backdoor, and Worm, the accuracy rates of this method are 89.21%, 87.48%, 87.21%, and 85.48%, respectively, which are 20.57%, 21.24%, 20.19%, and 30.85% higher than those of other methods. It can be proved that the system performance of this method is excellent and innovative.
6. Conclusion

Aiming at the problems of network intrusion detection in the era of big data, this paper proposes an intrusion detection platform based on the ACGAN model in big data environment. Firstly, this paper proposes an ACGAN sample enhancement framework based on a two-dimensional convolution structure and introduces self-attention mechanism to expand the data volume of attack samples. The improved ACGAN introduces self-attention mechanism in G and D to fully extract the global features of the samples, so as to improve the generation quality of the model. At the same time, the gradient penalty mechanism is introduced to overcome the problems of gradient disappearance to speed up the convergence of the model. The experimental results show that the intrusion detection model in this paper has achieved good detection accuracy, false-positive rate, and false-negative rate. At present, the model is only tested in the dataset, and the effect is good. It also needs to be tested in the actual network environment to verify the real performance of the model in the actual network environment.

Data Availability

The labeled dataset used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest.

Acknowledgments

This study was supported by Wenhua University.

References

[1] K. Nayak, D. Marino, P. Efstathopoulos, and T. Dumitraș, “Some vulnerabilities are different than others,” in Proceedings of the International Workshop on Recent Advances in Intrusion Detection, pp. 426–446, Springer, Switzerland, Europe, September 2014.

[2] A. Mishra, K. Nadkarni, and A. Patcha, “Intrusion detection in wireless ad hoc networks,” IEEE Wireless Communications, vol. 11, no. 1, pp. 48–60, 2004.

[3] R. Gassais, N. Ezzati Jivan, J. M. Fernandez, and D. Aloise, “Multi-level host-based intrusion detection system for Internet of things,” Journal of Cloud Computing, vol. 9, no. 1, pp. 1–16, 2020.

[4] J. Kim, Y. Shin, and E. Choi, “An intrusion detection model based on a convolutional neural network,” Journal of Multimedia Information System, vol. 6, no. 4, pp. 165–172, 2019.

[5] A. Mohan, A. K. Singh, B. Kumar, and R. Dwivedi, “Review on remote sensing methods for landslide detection using machine and deep learning,” Transactions on Emerging Telecommunications Technologies, vol. 32, no. 7, 2021.

[6] A. Javaid, Q. Niyaz, and W. Sun, “A deep learning approach for network intrusion detection system,” in Proceedings of the 9th EAI Inter-national Conference on Bio-inspired Information and Commu-nications Technologies (formerly BIONETICS), pp. 21–26, EAI, New York, NY, USA, December 2016.

[7] C. Yin, Y. Zhu, J. Fei, and X. He, “A deep learning approach for intrusion detection using recurrent neural network,” IEEE Access, vol. 5, pp. 21954–21961, 2017.
[8] F. Qu, J. Zhang, and Z. Shao, "An intrusion detection model based on deep belief network," in Proceedings of the 2017 VI International Conference on Network, Communication and Computing, pp. 97–101, Kunming, China, December 2017.

[9] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep-learning approach to network intrusion detection," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 41–50, 2018.

[10] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, "BAT: deep learning methods on network intrusion detection using NSL-KDD dataset," IEEE Access, vol. 8, pp. 29575–29585, 2020.

[11] R. Ahmad and I. Alsmadi, "Machine learning approaches to IoT security: a systematic literature review," Internet of Things, vol. 14, Article ID 100365, 2021.

[12] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, "Generative adversarial networks," ACM Computing Surveys, vol. 54, no. 6, pp. 1–38, 2021.

[13] L. Jin, F. Tan, and S. Jiang, "Generative adversarial network technologies and applications in computer vision," Computational Intelligence and Neuroscience, vol. 2020, Article ID 1459107, 18 pages, 2020.

[14] J. Y. Kim, S. J. Bu, and S. B. Cho, "Malware detection using deep transferred generative adversarial networks," in Proceedings of the 24th International Conference on Neural Information Processing, pp. 556–564, Springer, Guangzhou China, October 2017.

[15] J.-Y. Kim, S.-J. Bu, and S.-B. Cho, "Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders," Information Sciences, vol. 460-461, no. 9, pp. 83–102, 2018.

[16] C. Yin, Y. Zhu, and S. Liu, "An enhancing framework for botnet detection using generative adversarial networks," in Proceedings of the 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD), pp. 228–234, Chengdu, China, May 2018.

[17] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection," Information Sciences, vol. 479, pp. 448–455, 2019.

[18] Y. Liang and H. Yao, "Research on GAN-based container code images generation method," in Proceedings of the 2020 19th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCAES), pp. 198–201, IEEE, Xuzhou, China, October 2020.

[19] M. Ngxande, J.-R. Tapamo, and M. Burke, "Bias remediation in driver drowsiness detection systems using generative adversarial networks," IEEE Access, vol. 8, pp. 55592–55601, 2020.

[20] S. Arora and P. Anand, "Binary butterfly optimization approaches for feature selection," Expert Systems with Applications, vol. 116, pp. 147–160, 2019.

[21] A. B. Shaik and S. Srinivasan, "A brief survey on random forest ensembles in classification model," in Proceedings of the International Conference on Innovative Computing and Communications, pp. 253–260, Springer, Ostrava, Czech Republic, March 2019.

[22] R. Abdulhammed, M. Faezipour, H. Musafer, and Abuzeinid, "Efficient network intrusion detection using pca-based dimensionality reduction of features," in Proceedings of the 2019 International symposium on networks, computers and communications (ISNCC), pp. 1–6, IEEE, Istanbul, Turkey, June 2019.

[23] R. Cheng and X. Feng, "Method for predicting travel time of motor vehicle based on stack noise reduction self-encoder," Journal of Computers, vol. 31, no. 3, pp. 195–205, 2020.