Int. J. Appl. Math. Comput. Sci., 2024, Vol. 34, No. 2, 323–334
DOI: 10.61822/amcs-2024-0023

A RECOMBINATION GENERATIVE ADVERSARIAL NETWORK FOR INTRUSION DETECTION

HAOQI LUO [a], LIANG WAN [a,*]

[a] State Key Laboratory of Public Big Data
Guizhou University
Huaxi, Guiyang, 555025, PR China
e-mail: lwan@gzu.edu.cn

The imbalance and complexity of network traffic data are hot issues in the field of intrusion detection. To improve the detection rate of minority class attacks in network traffic, this paper presents a method for intrusion detection based on the recombination generative adversarial network (RGAN). In this study, dual-stage game learning is used to optimize the discriminator for efficient identification of attack samples. In the first stage, the proposed model trains a deep convolutional generative adversarial network (DCGAN) integrated with the self-attention (SA) mechanism, and simultaneously trains an independent convolutional neural network (CNN) classifier integrated with the gated recurrent unit (GRU). This stage allows the generator to generate minority class attack samples that closely resemble real samples, while the independent classifier possesses the basic classification ability. In the second stage, the generator and the independent classifier of the DCGAN together constitute the second layer of the model—the generative adversarial network. Through dual-stage game learning, the classifier's discrimination ability for the minority samples is optimized, and it serves as the final output of the discriminator. In addition, the introduction of reconstruction loss helps prevent the detection rate of false positive samples. Experimental results on the CSE-IDS-2018 dataset demonstrate that our model performs well compared with various other intrusion detection techniques in terms of detection accuracy, recall, and F1-score for minority class attacks.

Keywords: intrusion detection, generative adversarial network, class imbalance, RGAN.

1. Introduction

With the expansion of the Internet, a growing amount of sensitive information is being uploaded onto the network. Alongside this, new and intricate intrusion behaviors continue to emerge. Consequently, research on network-based intrusion detection holds significant importance (Sabahi and Movaghar, 2008). Network-based intrusion detection systems (NIDSs) play a crucial role in maintaining network security by distinguishing between legitimate and malicious network activities. However, in real-world networks, the proportion of malicious traffic is relatively small, resulting in a notable class imbalance within most intrusion detection datasets. This imbalance creates difficulties in detecting minority class attacks, thereby posing a threat to network security (Bedi *et al.*, 2021).

In recent years, there has been observed significant research and practical implementation of deep learning in the field of network intrusion detection (Kumar *et al.*, 2021; Wang *et al.*, 2018; Kanna and Santhi, 2021; Liu *et al.*, 2022; Thakkar and Lohiya, 2023). Deep learning-based intrusion detection models construct classification models that learn from training sets to identify network attack traffic. However, deep learning approaches typically require a large amount of training data, which poses a challenge in effectively detecting a small number of attack samples (Gupta *et al.*, 2022). Traditional methods attempt to address this issue by either undersampling the majority class samples or oversampling the minority class samples. However, these methods often result in overfitting or introduce noise into the dataset (Oksuz *et al.*, 2021).

The application of generative adversarial networks (GANs) has shown promise in addressing the class imbalance issue in network intrusion detection, as recommended by Goodfellow *et al.* (2014). GANs are utilized to simulate the distribution of authentic data and

[*] Corresponding author

generate synthetic samples for data augmentation (Bedi *et al*., 2021). Through adversarial learning, the generator within the GAN aims to generate pseudo-samples closely resembling real samples, while the discriminator develops a strong ability to distinguish real samples (Jabbar *et al*., 2021). Based on this concept, we propose a method for detecting minority class attacks using a recombination generative adversarial network. In this approach, the generative adversarial network utilizes the minority class samples as the real samples for game training. By doing this, the discriminator can learn better and identify the minority class samples. The contributions of this paper can be summarized as follows.

(i) A new intrusion detection method based on a two-layer improved generative adversarial network is proposed to accurately identify minority class attacks in network traffic.

(ii) For the purpose of improving the generator's ability to generate pseudo-samples, a self-attention mechanism is incorporated, and the classification capability of the discriminator is enhanced by integrating GRU.

(iii) By introducing the transfer learning mechanism and reconstruction loss function, the detection ability of the detector to abnormal samples is improved.

The rest of the paper is organized as follows. Section 2 introduces the relevant studies and prior work related to the proposed method. Section 3 elaborates on our suggested methodology. We evaluate the performance and effectiveness of the model through simulations in Section 4. Section 5 concludes the paper.

2. Related works

In this section, a variety of concerns regarding issues covered in this paper are discussed, including network intrusion detection and class imbalance in intrusion detection.

2.1. Network intrusion detection. Xiao *et al*. (2019) suggested a network intrusion detection model, the CNN-IDS, based on convolutional neural networks. Zhou *et al*. (2020) introduced a new intrusion detection framework that combines feature selection, ensemble learning techniques, and voting technology to augment the performance of the intrusion detection model. The framework first selects optimal feature subsets based on the connection between features. Then, by leveraging the voting technology and the probability distribution of basic learners, the framework is able to effectively identify attacks. Through this approach, the general performance of the intrusion detection model is improved, offering better accuracy and reliability in detecting and mitigating potential intrusions. Liao *et al*. (2022) preprocess traffic through filtering and gray level conversion to realize traffic visualization, as well as analyze and cluster the gray level of traffic to detect network attacks more accurately. Laghrissi *et al*. (2021) adopted mutual information (MI) and principal component analysis (PCA) as reduction and feature selection techniques to implement a deep learning approach based on long short-term memory (LSTM) to detect attacks.

Brunner *et al*. (2022) applied the stacked integration and tree structure Parzen estimator hyperparameter optimization method composed of a spiking neural network (SNN) and autoencoder (AE) models to intrusion detection, and the consequences showed that this way could enhance the performance of existing models. Qazi *et al*. (2023) built a hybrid intrusion detection system founded on deep learning that uses convolutional neural networks for convolution to collect local features, while deep recurrent neural networks extract features, improving the efficiency and predictability of the intrusion detection system. Wang *et al*. (2021) designed a combined deep intrusion detection framework based on SDAE-ELM. Their model effectively addresses the challenges commonly encountered in current deep neural network models, containing long training times and low detection accuracy. Additionally, their model enables a timely response to such behaviors, optimizing the overall proficiency of the intrusion detection system. Zou *et al*. (2023) recommended a network intrusion detection strategy, HC-DTTWSVM, based on decision tree double hierarchical clustering and a support vector machine, which can proficiently detect various categories of network intrusion and has good efficiency of detection.

2.2. Class imbalance in intrusion detection. We mainly introduce the recent research on the issue of class imbalance in intrusion detection. Zhang *et al*. (2022a) employed adaptive synthetic sampling (ADASYN) and random undersampling approaches to tackle the issue of data imbalance in intrusion detection. They found that LightGBM, a gradient boosting framework, excelled the other models in terms of its ability to handle data imbalance and detect intrusions accurately. Gupta *et al*. (2022) presented an intrusion detection method utilizing cost-sensitive and integrated algorithms, which assigns weights to different samples to decrease the false positive rate and enhance the data balance. Sun *et al*. (2023) introduced a low-lens intrusion detection approach with the attention mechanism based on a prototype capsule network, which is superior to the most advanced methods on unbalanced data sets. Andresini *et al*. (2021) trained generative adversarial networks (GANs) for data enhancement by representing network traffic as 2D images, and then trained CNN-based intrusion

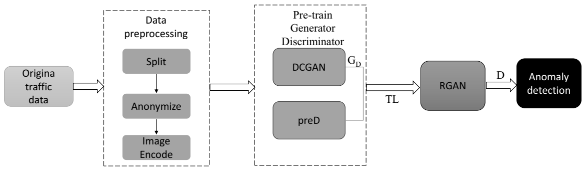


Fig. 1. Architecture of the proposed method.

detection models. The proposed method has better prediction accuracy on four benchmark data sets. Yuan *et al*. (2023) introduced a data balancing technique named the B-GAN. They utilized LSTM networks in both the generator and discriminator to enhance the understanding of data patterns and generate high-quality anomaly samples. This approach aimed to augment the capability of intrusion detection models in identifying intrusions accurately. Fu *et al*. (2021) introduced an intrusion detection data generation approach using generative adversarial networks to address the shortage of intrusion detection data and the sluggish updating process of mainstream detection methods. This method aimed to generate synthetic data samples that can be used to enhance the training process and optimize the performance of intrusion detection systems. Cui *et al*. (2023) proposed a novel multi-module merged intrusion detection system (IDS) called the GMM-WGAN-IDS. The system alleviates the influence of class unbalance problem through three parts: feature extraction, unbalance treatment and classification.

3. Proposed method

The main concept of a generative adversarial network is to train a generator to produce pseudo-samples that closely resemble real samples, while simultaneously training a discriminator to differentiate between real and pseudo-samples. Through an adversarial game mechanism, the discriminator becomes proficient at identifying real samples. According to this theory, the GAN's discriminator can effectively detect minority class attack samples by treating the minority class as the real sample Zhang *et al*. (2022b).

To address the problem of class imbalance in intrusion detection, the paper proposes a model called the recombination generative adversarial network (RGAN) for minority class attack intrusion detection. The architecture of this method is depicted in Fig. 1. The data are preprocessed to match the input format required by the neural network. Subsequently, an optimized deep convolutional adversarial generation network (DCGAN) generator is trained, which incorporates a self-attention (SA) module. An independent initial discriminator is integrated using the gated recurrent unit (GRU) and a convolutional neural network (CNN). These structures and parameters are reorganized into a new generative adversarial network called the RGAN using transfer learning. After another round of adversarial training, the discriminator of the RGAN serves as an anomaly detector.

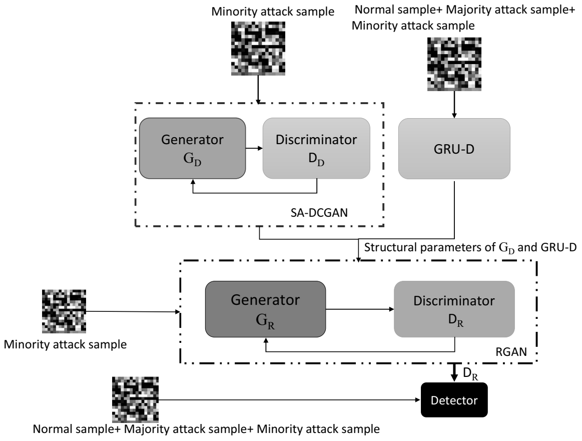

Minority attack sample
Normal sample+ Majority attack sample+ Minority attack sample
Generator G_D
Discriminator D_D
SA-DCGAN
GRU-D
Structural parameters of G_D and GRU-D
Generator G_R
Discriminator D_R
Minority attack sample
RGAN
D_R
Detector
Normal sample+ Majority attack sample+ Minority attack sample

Fig. 2. Specific framework of the RGAN-IDS.

In Fig. 2, the minority class samples are used as the training data for the DCGAN, which results in a generator with an excellent generation ability. Once the adversarial learning process is completed, the discriminator D_D is discarded. Secondly, the complete dataset is used to train the initial classifier GRU-D, allowing it to acquire a basic classification ability. Then, the architecture and hyperparameters of the generator G_D and GRU-D are input into the recombination generative adversarial network, and the minority samples are once again used for adversarial training to optimize the classifier's ability to differentiate between them. Finally, the classifier serves as an anomaly detector, improving the detection ability of a few attack samples.

3.1. Data preprocessing. In this paper, we select to use the relatively new CICIDS2018 dataset Pcap file with many attack types instead of the processed CSV file. This dataset is simulated and generated based on the real network traffic distribution, making it suitable for conducting research on network intrusion detection. It includes a substantial amount of original traffic. The format of the Pcap file is shown in Fig. 3. The packet header contains captured packets, each of which is divided into a packet header and packet data. The raw data flow in the Pcap format undergoes standardization to align with the input format of a neural network. This processing procedure consists of anonymization and digital encoding, ensuring the data conforms to the required format for neural network input.

Network granularity influences the analysis of data structures and distributions. Dainotti *et al*. (2012) provided a summary of the different levels of traffic

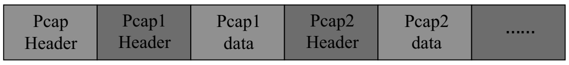


Fig. 3. Format of Pcap files.

granularity typically utilized in network data flow analysis. These include TCP connections, flows, sessions, services, and hosts. In this paper, we chose to anonymize the quintuple extracted from Pcap files to create session samples for detection analysis. These quintuples are important indicators used for feature extraction and differentiation in our research. Intrusive and normal data packets may have different patterns or values in these features, making them the inputs to our deep neural network model.

To protect data privacy and avoid providing useful clues from address information, we decided to anonymize the addresses. Specifically, we replaced the MAC addresses with 0:00:00:00:00:00:00 and the IP addresses with 0.0.0.0. By anonymizing the MAC addresses to a uniform value and replacing all IP addresses with the same anonymous value, we eliminated the variability and usage of these address information in the deep neural network model. This ensures that the model does not rely on specific MAC or IP addresses for classification. Through this anonymization process, we can safeguard the privacy of the data and remove any potential identification from the session samples. This allows us to focus on the analysis of other features and patterns present in the dataset, without compromising the security and privacy of the original addresses.

Network traffic can be viewed as a byte stream, where each byte has a data range from 0 to 255, similar to the range of grayscale values in an image. Thus, we can convert each byte into a grayscale pixel ranging from 0 to 255, which serves as input for our deep learning model. This transformation process effectively preserves the information embedded in the original data.

By adopting this approach, we can convert the anonymized session samples into input data suitable for deep learning models while protecting privacy and accurately preserving the original data's information. This conversion enables us to leverage the power of deep learning techniques in analyzing network traffic data, ensuring the preservation of useful features while maintaining data privacy.

3.2. Training the initial generator and classifier.

3.2.1. Generator with the self-attention mechanism. The basic network of the deep convolutional generative adversarial network (DCGAN) (Radford *et al.*, 2016), including the generator and discriminator, is based on a convolutional neural network (CNN). The

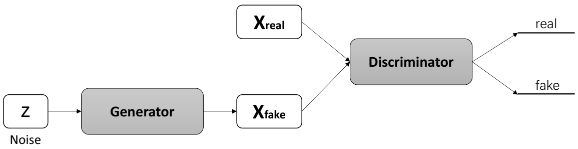


Fig. 4. Specific structure of the DCGAN.

CNN's powerful feature extraction ability is utilized to augment the learning effectiveness of the generative network. As shown in Fig. 4, X_{real} symbolizes the real sample, and X_{fake} symbolizes the generated fake sample. The generator learns the distribution of real samples and generates a new sample, X_{fake}, from the input random noise. The function of the discriminator is to differentiate genuine samples from artificially generated ones. Through continuous learning, the generator improves its ability to generate new samples. The performance of the DCGAN reaches its peak when the discriminator cannot precisely identify the genuineness of the input sample. The objective function of the DCGAN is

$$\begin{aligned}&\min_G \max_D V(D,G)\\&= E_{x\sim p_{\text{true}}(x)}[\log D(x)]\\&\quad + E_{z\sim p_z(z)}[\log(1-D(G(z)))].\end{aligned} \tag{1}$$

During the training process of a DCGAN, random noise (represented by p_z) works as input to the generator to generate fake samples. The objective for the discriminator is to correctly distinguish between real (X_{real}) and generated fake (X_{fake}) samples. The objective function $V(D,G)$ of the DCGAN seeks to optimize both the generator and discriminator simultaneously. When the DCGAN reaches the optimal point, known as the Nash equilibrium, the min-max problem of $V(D,G)$ is solved optimally. This equilibrium is achieved if and only if the underlying distributions of the original data (p_{true}) and the input noise (p_z) are identical. In this scenario, the generator can generate X_{fake} samples that closely resemble the real ones.

The features generated by the traditional DCGAN are only derived by the fixed spatial local information in the image, which can generate high-quality and detailed texture features, but the effect of capturing specific geometric features is not good. In this paper, the self-attention (SA) module is integrated with the DCGAN so that it can adapt to more related features and select useful information from that which is redundant. How this module works is shown in Fig. 5.

Note that x_i is the input to the generator, and the weight matrices to be learned are expressed as W_q, W_k and W_q. The obtained feature spaces Q_i, K_i, V_i result from the multiplication of the image features with distinct

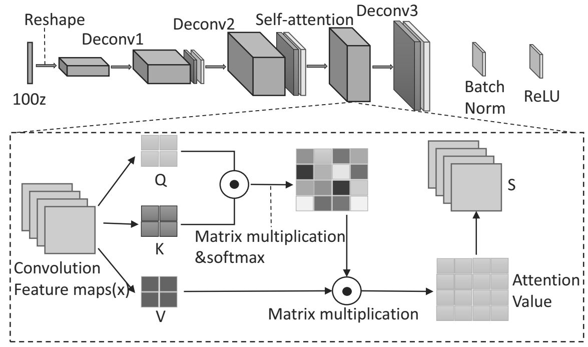


Fig. 5. Specific structure of the SA generator.

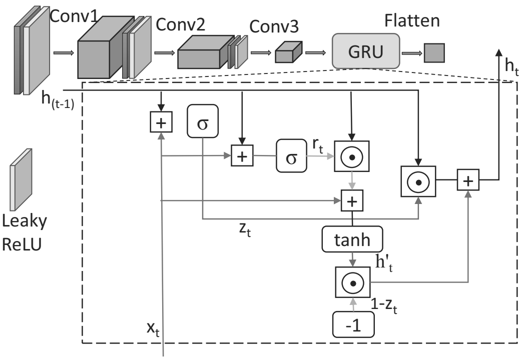


Fig. 6. Specific structure of GRU-D.

weight matrices. We multiply the transpose of Q_i by K_i to get C_{ij}. Then, C_{ij} is normalized with SoftMax to generate attention features. The calculation method is as follows:

$$Q_i = W_q x_i, \tag{2}$$

$$K_i = W_k x_i, \tag{3}$$

$$V_i = W_\nu x_i, \tag{4}$$

$$C_{ij} = Q_i^T K_j, \tag{5}$$

$$m_{ij} = \frac{\exp(C_{ij})}{\sum_{i=1}^{N} \exp(C_{ij})}. \tag{6}$$

Finally, the adaptive attention feature map $h = (h_1, h_2, \dots, h_j, \dots, h_N)$ is obtained by multiplying the attention map m_{ij} with V_i and h_j is calculated with the following formula:

$$h_j = \sum_{i=1}^{N} m_{ij} V_i, \tag{7}$$

$$S_j = ro_j + x_i. \tag{8}$$

By integrating all spatial information and local information, S_j is obtained. In order to account for the correlation between neighborhood information and long-distance features, a transition parameter r is introduced. It initially starts with the zero value and gradually increases. This mechanism is utilized to assign weights to other long-distance feature details, allowing a better representation of the relationship between neighborhood information and distant features. The calculation formula is expressed by Eqn. (8).

3.2.2. Generator with the self-attention mechanism.

GRU-D is a binary classification model that can be trained and transferred to the discriminator of the RGAN. Because generators in the RGAN are transferred from DCGAN generators, GRU-D uses the same discriminator structure as the DCGAN, making generators and discriminators compatible. The gate recurrent unit (GRU) is a version of the RNN, close to long short-term memory (LSTM), which is utilized in overcoming the challenge of gradient disappearance in RNNs during training (Nosouhian *et al.*, 2021). GRU is simpler and more efficient than LSTM. Because the training set data has a certain time dependence, this paper combines the GRU and CNN to transform the discriminator model of the DCGAN into a model composed of the GRU network, so that it can recognize continuous data better. The improved GRU discriminator has a basic structure similar to the original discriminator of the DCGAN. After the convolutional layer, GRU units are used to replace the original fully connected layer. The final discriminator structure is shown in Fig. 6.

The GRU determines the degree of neuron information retention and forgetting by controlling the update gate and reset gate, and can learn the long and short time series features. The diagram provided in Fig. 6 illustrates the network structure, and the formula of the update gate is as follows:

$$r_t = \sigma(w_{rx} x_t + w_{rh} h_{t-1} + b_r), \tag{9}$$

$$z_t = \sigma(w_{zx} x_t + w_{zh} h_{t-1} + b_z), \tag{10}$$

where $h_{(t-1)}$ and x_t are respectively the hidden layer output of the previous time and the input of the current time, σ is the sigmoid activation function, w_{rx} and w_{rh} are the weights of input and hidden states, respectively, w_{zx} and w_{zh} are the weights of the input and hidden states, and b_r and b_z are the offset of the reset door and the update door.

The candidate hidden state is utilized to support the generation of the final hidden state output; the reset gate determines whether the hidden state from the preceding time step influences the candidate hidden state at the current time step, while the update gate is responsible for updating the candidate hidden state. According to Eqn. (12), if the numerical value assigned to the update gate is close to 1 in t_2 and t_2 ($t_1 < t_2$), the hidden state hardly flows into the current hidden state during this period, and the hidden state information before t_1 to t_2 is updated, which solves the difficulty of gradient attenuation of the recurrent neural network. As a result, the GRUs are better able to capture dependencies for features with larger time step distances,

$$\tilde{h}_t = \tanh(w_{\widehat{hx}} x_t + w_{\widehat{hh}}(r_t \odot h_{t-1}) + b_{\widehat{h}}), \tag{11}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t. \tag{12}$$

In the formula, $w_{\widehat{hx}}$ and $w_{\widehat{hh}}$ are the weight matrices of candidate vectors in the input and hidden layers, respectively, $\tilde{h}_t$ is the output of the candidate hidden state, h_t is the output of the current hidden layer, and $\odot$ is the Hadamard product, which refers to the multiplication of the corresponding elements.

3.3. Recombination generate adversarial networks. The recombination generative adversarial network (RGAN) is composed of the generator from the pre-trained DCGAN and GRU-D. The structure details of the RGAN architecture are shown in Fig. 7.

We choose ReLU normalization in the RGAN because the ReLU activation function has the following advantages; ReLU is a nonlinear activation function that introduces nonlinearity and enhances the expressive power of the model. This is crucial for GANs since their goal is to learn complex data distributions. Furthermore, the derivative of the ReLU function in the positive range is 1, which means the gradients are not affected by extremely small gradient values, thus alleviating the problem of gradient vanishing. In comparison with ReLU, $\tanh$ normalization has some differences. For example, the output range of the $\tanh$ function is $[-1, 1]$, while the output range of the ReLU function is $[0, +\infty]$. This leads to differences in the range of the mean and variance used in batch normalization. Additionally, the derivative of the $\tanh$ function tends to approach zero for inputs larger than 2 or smaller than -2, causing the problem of gradient vanishing. On the other hand, the derivative of ReLU in the positive range is always 1, which helps to address the issue of gradient vanishing. In summary, we choose ReLU normalization to introduce nonlinearity and simplify the problem of gradient vanishing. The $\tanh$ normalization may face the problem of gradient vanishing and needs to consider the difference in the output range.

When RGANs undergo adversarial training, the generated pseudo-samples become closer to the real sample, and this could potentially result in an increase in false positive rates. Hence, this research paper suggests incorporating a reconstruction loss component into the objective function of the RGAN. This addition aims to enhance the discriminator's capacity to differentiate real samples (normal samples) and subsequently decrease the occurrence of false positives.

The generator loss and the discriminator function can respectively be expressed as

$$E_{z\sim P_z(z)}[\log(1 - D(G(Z)))], \tag{13}$$

$$E_{x\sim p_{\text{data}}(x)}[\log D(x)] + E_{x\sim p_z(z)}[\log(1 - D(G(Z)))]. \tag{14}$$

The representation of the new loss function, incorporating reconstruction loss, can be as follows:

generator:

$$L_G^u + \theta \times L_X^u, \tag{15}$$

discriminator:

$$L_D^v + \theta \times L_X^v. \tag{16}$$

where L_G^u and L_X^u represent respectively, the loss function of the initial generator and discriminator. The reconstruction loss is controlled by parameter θ. Upon training the discriminator, the generator's parameter θ is fixed and does not participate in training, and the same is true for the discriminator's parameter v upon training the generator. Equation (15) indicates that the generator receives two source inputs of discriminator classification results and real data L_1 reconstruction losses. By introducing reconstruction losses, the generator is capable of producing samples by utilizing the features or information provided by the discriminator, thus furnishing extra details to the discriminator. Therefore, the discriminator's capacity to differentiate genuine samples will be enhanced further.

4. Experiments

In this section, we first describe the dataset and evaluation indicators of our simulations, and then the experimental results are evaluated and compared.

4.1. Dataset and evaluation indicators. The dataset utilized in this paper is comparatively new in the domain of intrusion detection and is called CSE-CIC-IDS2018. It is produced through the simulation of an actual network traffic distribution and has obvious imbalance, which is very consistent with our research problem. By sampling the majority class and minority class samples in the original dataset, the dataset used in this paper is shown in Table 1.

To assess the effectiveness of the suggested approach, we used the following evaluation indicators: accuracy, precision and F1-score. Each index can be represented by four quantities, that is true positive (TP), false positive (FP), true negative (TN), false negative (FN). TP represents the number of positive samples accurately detected as positive; FP signifies the count

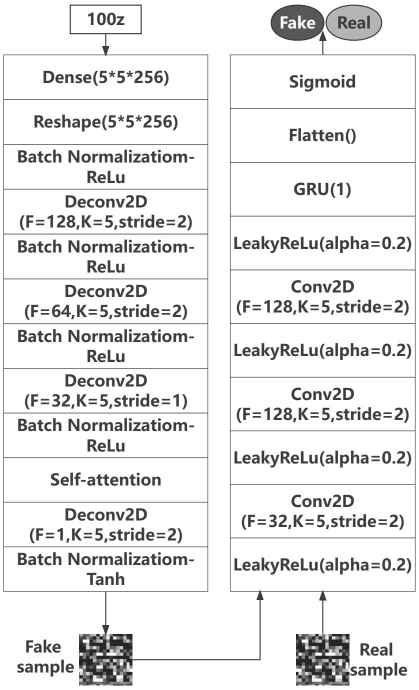


Fig. 7. Structural parameters of the RGAN. "$100z$" represents a 100-dimensional random noise vector from a specific distribution. The convolutional layers indicate the number of filters (F), kernel size (K), and stride size in brackets. The dense layers indicate the number of neurons in brackets.

of positive samples wrongly recognized as negative; TN indicates the amount of negative samples rightly detected; FN symbolizes the count of negative samples mistakenly detected as positive.

Accuracy indicates the proportion of samples that are accurately identified, and its calculation formula is

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}. \tag{17}$$

Precision is the percentage of the count of predicted samples belonging to a category to the overall amount of such samples:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{18}$$

Recall is the ratio of the total of samples rightly detected as a category to the overall number of samples in that category. In intrusion detection, the recall rate of the calculated attack class can also be called the attack detection rate. In this study, the detection rates of normal samples and attack samples are calculated as follows:

$$\text{DR}_{\text{normal}} = \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{19}$$

Table 1. Dataset of experiments.

Attack types	Number of samples	Proportion
Benign	71799	44.16%
DDoS attacks-LOIC-HTTP	42000	25.83%
DoS-Goldeneye	29550	18.17%
Bot	13980	8.59%
Brute-Force-Web	3159	1.94%
Brute-Force-XSS	1367	0.84%
Infiltration	452	0.27%
Sql-Injection	276	0.16%

$$\text{DR}_{\text{attack}} = \frac{\text{TN}}{\text{TN} + \text{FP}}. \tag{20}$$

F1-score can be expressed as the harmonic average of the precision and recall values, which assigns equal weights to the precision and the recall scores, and is an evaluation index. It can be calculated according to

$$\text{F1-score} = \frac{2}{\dfrac{1}{\text{recall}} + \dfrac{1}{\text{precision}}}. \tag{21}$$

We also use two evaluation indexes macro-mean and weighted-mean, to measure the macroscopic classification effect of the intrusion detection model. The formula for calculating macro-means (macro-mean) is as follows:

$$\text{macro-mean-}R = \frac{1}{m}\sum_{i=1}^{1} R_i. \tag{22}$$

Since the weighted-mean is a percentage of the sample size, it can only reflect the detection effect of most types of samples. To represent the weights of a few class samples, we use the improved weights to calculate the parameter weighted mean. The following Equation (23) is shown for taking precision as an example, where m represents the number of samples, β_i stands for the proportion between the amount of samples of a specific category and the overall count of samples in a classification task, and R_i refers to the recall, which signifies the capacity of a model to correctly identify positive instances of that particular class out of all actual positive instances:

$$\text{weighted-mean-}R = \sum_{i=1}^{m} \frac{1-\beta_i}{\sum_{j=1}^{m}(1-\beta_j)} R_i. \tag{23}$$

4.2. Determining the size of session samples.

In order to confirm the effect of the reconstruction loss function on the detection of normal samples by the discriminator of the reconstructed adversarial network, we set the values of parameter θ as 0, 0.25, 0.5, 0.75, 1, 5, 10. By comparing accuracy, macro-mean-F1 and weighted-mean-F1, θ values with high scores are selected. The results are shown in Fig. 8. The horizontal axis is θ value, and the vertical axis exhibits the three appeal values, which are typically color-coded as squares, circles and triangles.

As can be seen from Fig. 8, when θ is 1, the three evaluation indexes of the model attain their maxima, i.e., 0.9977, 0.9430 and 0.9330, respectively. This means that, when $\theta = 1$, the model has the best detection performance. Therefore, we choose 1 as the reconstruction parameter.

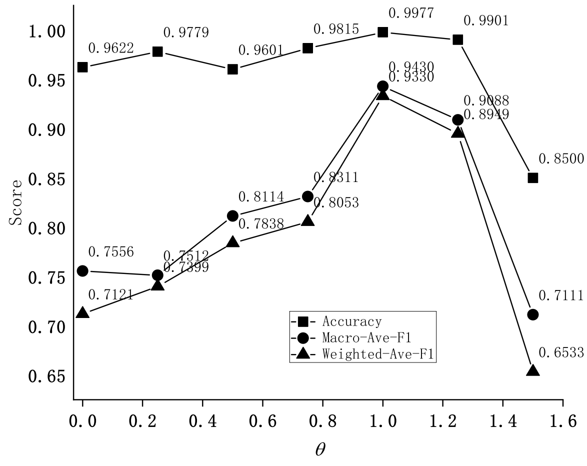
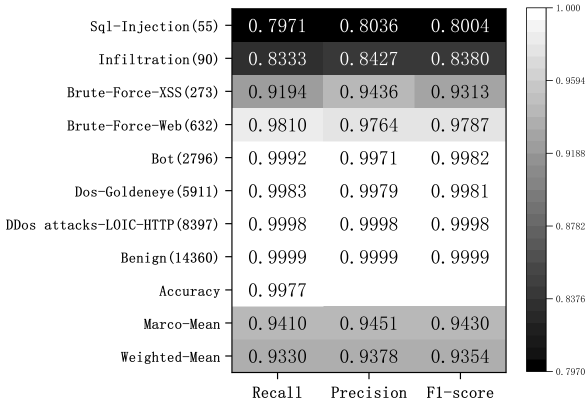


Fig. 8. Accuracy, macro-mean-F1 and weighted-mean-F1 of θ.

4.3. Evaluating the proposed method.

As introduced earlier, our experimental dataset is a partial category of CSE-CIC-IDS2018. Figure 9 displays the detection result as a heat map.



	Recall	Precision	F1-score
Sql-Injection(55)	0.7971	0.8036	0.8004
Infiltration(90)	0.8333	0.8427	0.8380
Brute-Force-XSS(273)	0.9194	0.9436	0.9313
Brute-Force-Web(632)	0.9810	0.9764	0.9787
Bot(2796)	0.9992	0.9971	0.9982
Dos-Goldeneye(5911)	0.9983	0.9979	0.9981
DDos attacks-LOIC-HTTP(8397)	0.9998	0.9998	0.9998
Benign(14360)	0.9999	0.9999	0.9999
Accuracy	0.9977		
Marco-Mean	0.9410	0.9451	0.9430
Weighted-Mean	0.9330	0.9378	0.9354

Fig. 9. Classification result of the proposed method.

Figure 9 shows the F1-score, recall, precision and accuracy of various samples. Also shown below are whole accuracy, macro-mean and weighted-mean of recall, precision and F1-score. The color in the heat map changes according to the value, as indicated by the vertical bar on the right side of Fig. 9. We show the values of each sample of the test set in parentheses. As can be seen from Fig. 9, the suggested method exhibits good performance. This approach ensures not only efficient detection for the majority samples, but also enhances the detection rate for minority samples. Therefore, this method is efficient for the class imbalance problem in intrusion detection.

In addition, we also evaluated the effect of adding SA and GRU modules to verify their validity. Figure 10 shows the results of ablation experiments using accuracy, weighted-mean of precision, recall and F1-score as evaluation indexes. It can be seen that the generator of the original DCGAN and the basic CNN classifier have the lowest scores out of the four evaluation indicators. After the integration of the SA mechanism and GRU module, the four values are increased by about 4.56%, 5.93%, 6.86% and 6.40%, respectively to attain optimal detection performance.

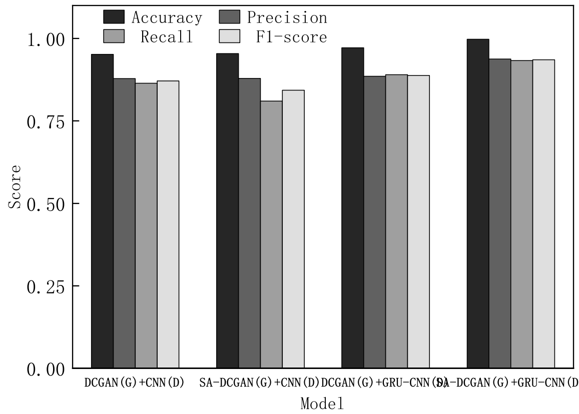


Fig. 10. Ablation evaluation of the proposed method.

4.4. Comparison.

In this section, we execute a comparative analysis between the suggested methods and various conventional intrusion detection techniques. Figures 11–13 shows the comparison of the proposed framework with five common machine learning algorithms including a dynamic neural network (DNN), a convolutional neural network (CNN), XGBoost, a support vector machine (SVM) and random forest (RF) in terms of weighted mean parameters. It can be seen that, compared with the conventional method, the suggested framework is improved in each index and shows better detection performance.

Additionally, since this paper is motivated by the work of Zhang *et al.* (2022b) and builds upon the method, our model is compared with the method proposed by Zhang *et al.* (2022b) to confirm the performance of our

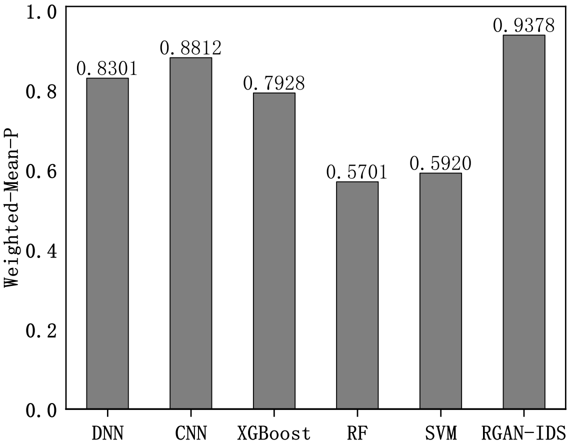


Fig. 11. Comparison of weighted-mean-precision with traditional machine learning.

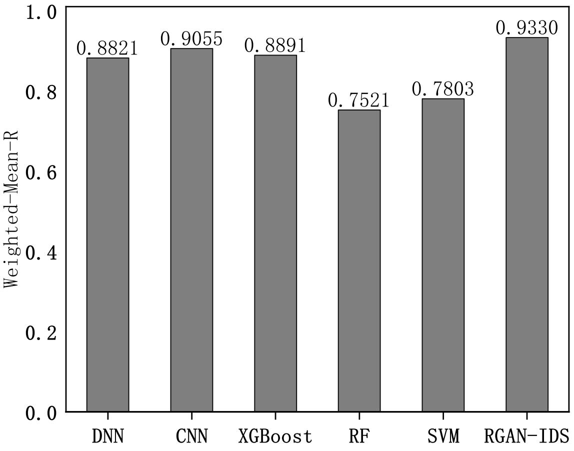


Fig. 12. Comparison of weighted-mean-recall with traditional machine learning.

upgrade. As can be seen in Fig. 14, our method has more effective performance on a macrolevel, and accuracy, macro-mean and weighted-mean are all improved, so that the comprehensive detection effect of our model is better.

In order to prove that our model can boost the detection rate of the minority sample, we also compare the detection indexes of the two methods for a few classes, Brute-Force-Web, Brute-Force-XSS, Infiltration and Sql-Injection, on the same data set. Figures 15–17 show the results. Figures demonstrate that our approach exhibits favorable detection results for several sample types present in the dataset. Our method outperformed the TGAN-IDS in all three evaluation metrics, indicating a significant and noticeable difference.

We also compare our approach with several class imbalanced intrusion detection models (Bedi *et al.*, 2021; Cui *et al.*, 2023; Gelenbe and Nakip, 2023) on the same dataset. Weighted-mean of precision, recall and F1-score are used as evaluation metrics, and the results obtained from the experiment are presented in Table 2.

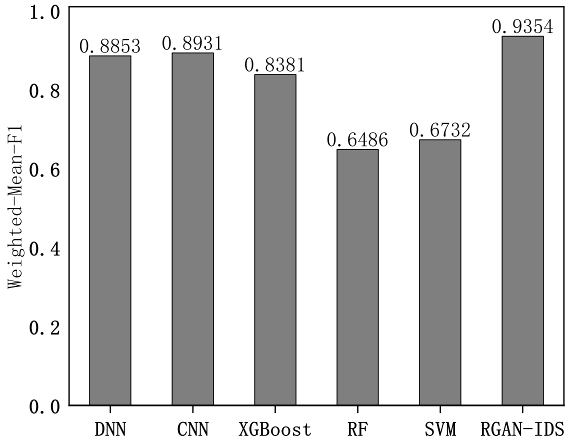


Fig. 13. Comparison of weighted-mean-F1-score with traditional machine learning.

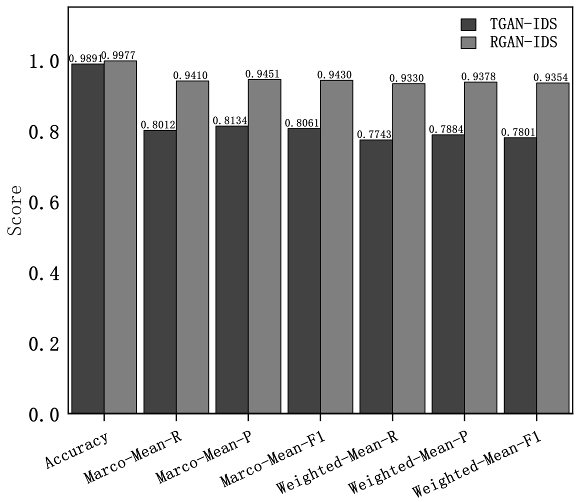


Fig. 14. Comparison of macro-index with the TGAN-IDS.

From the data presented in the table, it is evident that the three indexes of the RGAN-IDS outperform both the I-Siam-IDS and GMM-WGAN-IDS. Specifically, when compared with the I-Siam-IDS, the weighted-mean of precision, recall, and F1-score exceed 9.57%, 7.22%, and 8.4%, respectively. Similarly, when compared with the GMM-WGAN-IDS, the three indexes exceed 3.9%, 2.3%, and 3.1%. In comparison with the recent successful research on the ARNN, although the precision is lower, the F1 score is higher. These findings indicate that our method effectively addresses the class imbalance difficulty in intrusion detection, resulting in better detection performance for minority attack samples.

In order to validate the versatility of our method,

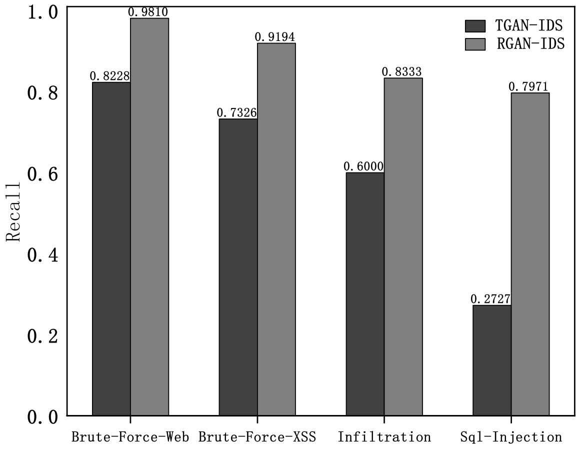


Fig. 15. Comparison of recall with the TGAN-IDS.

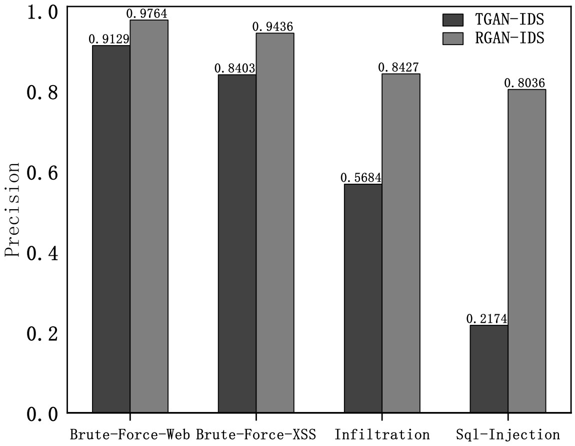


Fig. 16. Comparison of precision with the TGAN-IDS.

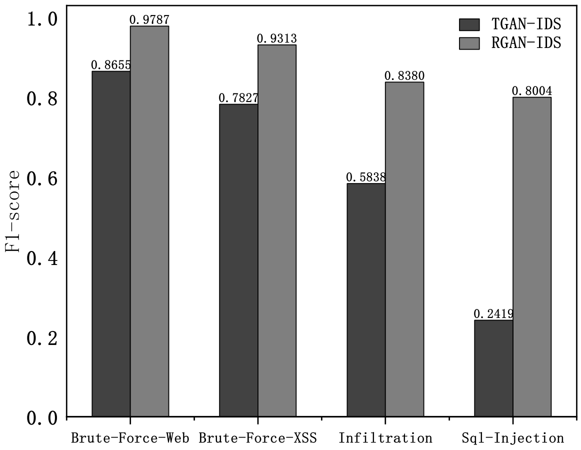


Fig. 17. Comparison of F1-score with the TGAN-IDS.

we performed tests on the UNSW-NB15 dataset. It is a widely used network intrusion detection dataset that contains network traffic data from real-world environments. By testing on this dataset, we were able to evaluate the performance of our method in real-world scenarios. According to the obtained results, the RGAN-IDS exhibits excellent performance in terms of precision, recall, and F1 score, with values of 98.11%, 98.95%, and 98.52% respectively. In comparison with these results, the I-Siam-IDS, GMM-WGAN-IDS, and ARNN have lower precision, recall, and F1 scores (Table 3).

Based on this testing, we were able to conclude that our method is not only effective on specific datasets, but also exhibits universality when applied to widely used datasets like UNSW-NB15. This demonstrates the effectiveness of our method and its ability to adapt to various network intrusion detection tasks.

To assess the efficiency of different models, we measured the training times of each model during our experiments. We trained multiple models using the same dataset and recorded the time taken for each model to complete the training process. According to Table 4, it can be observed that this method takes longer testing time compared with the other three methods. This could be due to the fact that this method involves the extraction of time features, while the other three methods may be more focused on other types of features or processing techniques. This may indicate that this method pays more attention to time-related data during processing, which may involve more calculations and processing steps, resulting in longer testing times. Although this method may require more time to complete testing, it may also have more advantages. Extracting time features may enable the model to better capture temporal changes and trends, thereby enhancing the model's predictive ability on time-series data. This method may be more suitable for problems where time is an important factor and time-related features have a significant impact on the data.

In conclusion, although this method may have longer testing time, its ability to extract time features may provide an advantage in solving time-related problems. The specific application needs to be evaluated based on the actual requirements.

5. Conclusions

The method proposed in this paper aims to optimize the detection rate of intrusion detection systems against minority attacks. It introduces an RGAN-IDS detection approach that utilizes a recombination generative adversarial network. The method combines a deep convolutional generative adversarial network (DCGAN) and self-attention (SA) mechanisms to optimize the generator. This optimization process enhances the generator's ability to generate strong pseudo-samples. Additionally, the RGAN is used to optimize discriminators that can efficiently distinguish

Table 2. Comparison with well-known ML models on CICIIDS2018.

	Precision [%]	Recall [%]	F1-score [%]
I-Siam-IDS	84.21	86.08	85.14
GMM-WGAN-IDS	89.88	91.00	90.44
ARNN	94.21	93.01	93.36
RGAN-IDS	93.78	93.30	93.54

Table 3. Comparison with well-known ML models on UNSW-NB15.

	Precision [%]	Recall [%]	F1-score [%]
I-Siam-IDS	89.32	83.51	86.31
GMM-WGAN-IDS	87.40	90.12	88.73
ARNN	97.33	98.53	97.94
RGAN-IDS	98.11	98.95	98.52

Table 4. Comparison of time complexity.

	Training time [s]	testing time [s]
I-Siam-IDS	1633	9.31
GMM-WGAN-IDS	1888	9.52
ARNN	857	8.13
RGAN-IDS	1790	9.15

between pseudo and real samples. Transfer learning techniques are applied to improve the performance of the RGAN. After adversarial training, the discriminator of the RGAN is used as the ultimate anomaly detector. To prevent a decline in the identification accuracy of normal instances during RGAN training, the loss function is reconstructed. Experimental results demonstrate that the RGAN-IDS effectively improves the anomaly detection capabilities for minority attacks. In the future, a further refinement of the model's structure is planned to enable the detection of unknown attacks as well.

Acknowledgment

The authors would like to thank Professor Wan Liang of the School of Computer Science, Guizhou University, for his guidance and support. Thanks also go to the providers of the public dataset CICIDS2018. This dataset is a valuable resource for our research and plays an important role in our work.

References

Andresini, G., Appice, A., De Rose, L. and Malerba, D. (2021). GAN augmentation to deal with imbalance in imaging-based intrusion detection, *Future Generation Computer Systems* **123**(2021): 108–127, DOI:10.1016/j.future.2021.04.017.

Bedi, P., Gupta, N. and Jindal, V. (2021). I-SIAMIDS: An improved SIAM-IDS for handling class imbalance in network-based intrusion detection systems, *Applied Intelligence* **51**(2): 1133–1151.

Brunner, C., Ko, A. and Fodor, S. (2022). An autoencoder-enhanced stacking neural network model for increasing the performance of intrusion detection, *Journal of Artificial Intelligence and Soft Computing Research* **12**(2): 149–163.

Cui, J., Zong, L., Xie, J. and Tang, M. (2023). A novel multi-module integrated intrusion detection system for high-dimensional imbalanced data, *Applied Intelligence* **53**(1): 272–288.

Dainotti, A., Pescape, A. and Claffy, K.C. (2012). Issues and future directions in traffic classification, *IEEE Network* **26**(1,SI): 35–40.

Fu, W., Qian, L. and Zhu, X. (2021). GAN-based intrusion detection data enhancement, *Proceedings of the 33rd Chinese Control and Decision Conference (CCDC 2021), Kunming, China*, pp. 2739–2744.

Gelenbe, E. and Nakip, M. (2023). IoT network cybersecurity assessment with the associated random neural network, *IEEE Access* **11**: 85501–85512, DOI: 10.1109/ACCESS.2023.3297977.

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014). Generative adversarial nets, *28th Conference on Advances in Neural Information Processing Systems (NIPS 2014), Montreal, Canada*, pp. 2672–2680.

Gupta, N., Jindal, V. and Bedi, P. (2022). CSE-IDS: Using cost-sensitive deep learning and ensemble algorithms to handle class imbalance in network-based intrusion detection systems, *Computers & Security* **112**(2022): 102499, DOI:10.1016/j.cose.2021.102499.

Jabbar, A., Li, X. and Omar, B. (2021). A survey on generative adversarial networks: Variants, applications, and training, *ACM Computing Surveys* **54**(8): 1–49.

Kanna, P.R. and Santhi, P. (2021). Unified deep learning approach for efficient intrusion detection system using integrated spatial-temporal features, *Knowledge-Based Systems* **226**: 107132.

Kumar, Y., Chouhan, L. and Subba, B. (2021). Deep learning techniques for anomaly based intrusion detection system: A survey, *in* S. Paul and J. Verma (Eds), *2021 International Conference on Computational Performance Evaluation (COMPE-2021), Shillong, India,* pp. 915–920.

Laghrissi, F., Douzi, S., Douzi, K. and Hssina, B. (2021). Intrusion detection systems using long short-term memory (LSTM), *Journal of Big Data* **8**(1): 65.

Liao, D., Zhou, R., Li, H., Zhang, M. and Chen, X. (2022). GE-IDS: An intrusion detection system based on grayscale and entropy, *Peer-to-Peer Networking and Applications* **15**(3): 1521–1534.

Liu, C., Antypenko, R., Sushko, I. and Zakharchenko, O. (2022). Intrusion detection system after data augmentation schemes based on the VAE and CVAE, *IEEE Transactions on Reliability* **71**(2): 1000–1010.

Nosouhian, S., Nosouhian, F. and Khoshouei, A.K. (2021). A review of recurrent neural network architecture for sequence learning: Comparison between LSTM and GRU, *Preprints.org*: 202107.0252, DOI: 10.20944/preprints202107.0252.v1.

Oksuz, K., Cam, B.C., Kalkan, S. and Akbas, E. (2021). Imbalance problems in object detection: A review, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **43**(10): 3388–3415.

Qazi, E.U.H., Faheem, M.H. and Zia, T. (2023). HDLNIDS: Hybrid deep-learning-based network intrusion detection system, *Applied Sciences* **13**(8): 4921.

Radford, A., Metz, L. and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks, *ArXiv:* 1511.06434.

Sabahi, F. and Movaghar, A. (2008). Intrusion detection: A survey, *2008 3rd International Conference on Systems and Networks Communications, Slema, Malta*, pp. 23–26, DOI: 10.1109/ICSNC.2008.44.

Sun, H., Wan, L., Liu, M. and Wang, B. (2023). Few-shot network intrusion detection based on prototypical capsule network with attention mechanism, *Plos ONE* **18**(4): e0284632.

Thakkar, A. and Lohiya, R. (2023). Fusion of statistical importance for feature selection in deep neural network-based intrusion detection system, *Information Fusion* **90**(2023): 353–363, DOI: 10.1016/j.inffus.2022.09.026.

Wang, W., Sheng, Y., wang, J., Zeng, X., Ye, X., Huang, Y. and Zhu, M. (2018). HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection, *IEEE Access* **6**(2018): 1792–1806, DOI: 10.1109/ACCESS.2017.2780250.

Wang, Z., Liu, Y., He, D. and Chan, S. (2021). Intrusion detection methods based on integrated deep learning model, *Computers & Security* **103**(2021): 102177.

Xiao, Y., Xing, C., Zhang, T. and Zhao, Z. (2019). An intrusion detection model based on feature reduction and convolutional neural networks, *IEEE Access* **7**: 42210–42219, DOI: 10.1109/ACCESS.2019.2904620.

Yuan, L., Yu, S., Yang, Z., Duan, M. and Li, K. (2023). A data balancing approach based on generative adversarial network, *Future Generation Computer Systems* **141**(2023): 768–776.

Zhang, H., Ge, L. and Wang, Z. (2022a). A high performance intrusion detection system using LightGBM based on oversampling and undersampling, *in* D. Huang *et al.* (Eds), *Intelligent Computing Theories and Application (ICIC 2022)*, Lecture Notes in Computer Science, Vol. 13393, Springer, Cham, pp. 638–652, DOI: 10.1007/978-3-031-13870-6_53.

Zhang, X., Wang, J. and Zhu, S. (2022b). Dual generative adversarial networks based unknown encryption ransomware attack detection, *IEEE Access* **10**(2021): 900–913, DOI: 10.1109/ACCESS.2021.3128024.

Zhou, Y., Cheng, G., Jiang, S. and Dai, M. (2020). Building an efficient intrusion detection system based on feature selection and ensemble classifier, *Computer Networks* **174**(2020): 107247, DOI: 10.1016/j.comnet.2020.107247.

Zou, L., Luo, X., Zhang, Y., Yang, X. and Wang, X. (2023). HC-DTTSVM: A network intrusion detection method based on decision tree twin support vector machine and hierarchical clustering, *IEEE Access* **11**(2023): 21404–21416, DOI: 10.1109/ACCESS.2023.3251354.

Haoqi Luo is an MS student at the School of Computer Science and Technology, Guizhou University. Her current research interests include network security and intrusion detection.

Liang Wan is a professor at the School of Computer and Science and Technology, Guizhou University. His research interests include computer software and theory, and information security.

Received: 4 December 2023
Revised: 1 March 2024
Accepted: 2 April 2024