Research Article

Jing Yang*

WSN intrusion detection method using improved spatiotemporal ResNet and GAN

https://doi.org/10.1515/comp-2024-0018
received November 29, 2023; accepted September 30, 2024

Abstract: A network intrusion detection method that integrates improved spatiotemporal residual network and generative adversarial network (GAN) in a big data environment is proposed to address the issues of poor feature extraction and significant impact from data imbalance in most existing intrusion detection methods. First, GANs are used for wireless sensor network data resampling to generate new sample sets, thereby overcoming the impact of data imbalance. Then, an improved spatiotemporal residual network model is designed, in which the spatial and temporal features of the data are extracted and fused through multi-scale one-dimensional convolution modules and gated loop unit modules, and identity maps are added based on the idea of residual networks to avoid network degradation and other issues. Finally, the resampled samples are input into the improved spatiotemporal residual network model to output the intrusion detection results of the network. Based on the NSL-KDD, UNSW-NB15, and CICIDS2017 datasets, experimental analysis is conducted on the proposed method. The results showed that its accuracy on the three datasets is 99.62, 83.98, and 99.86%, respectively, which are superior to other comparative methods.

Keywords: network intrusion detection, improved spatiotemporal residual network, data resampling, identity mapping, wireless sensor network

1 Introduction

With the increasing amount of data transmitted by network devices and communication protocols, attack methods targeting the Internet have become more complex and diverse, and network security issues have become increasingly severe. The current computer network is facing security threats such as denial of service (DoS), viruses, and network sniffing. Intrusion detection systems (IDS) have become a research hotspot in network security protection technology [1–3].

Network intrusion detection (NID) is to determine whether a system has abnormal behavior based on network traffic data, and intrusion detection can also be abstracted as a classification problem. Early IDS mostly used manual extraction or machine learning (ML) algorithms for data feature extraction. With the advent of the big data era, network data is growing rapidly, and data features often have complex, high-dimensional, nonlinear, and other characteristics. Faced with complex data, detection engines that use traditional data feature extraction methods are no longer able to complete the data feature extraction task well, and there is a problem of poor data feature extraction ability [4,5]. In addition, when encountering network data with imbalanced data distribution, most traditional classification algorithms have higher accuracy in detecting normal samples that make up a large proportion of the overall sample size but lower accuracy in detecting abnormal samples with a small number, resulting in false positives and missed detections.

Deep learning (DL) algorithms are more efficient in model construction and feature extraction and have higher accuracy in processing large-scale data. Their powerful properties such as adaptive feature learning, nonlinear modeling ability, hierarchical abstract representation, and large-scale parallel computing make them an effective tool for intrusion detection [6,7]. Therefore, based on the application research of DL algorithms, an NID method is proposed that integrates improved spatiotemporal residual networks and generative adversarial networks (GANs) in a big data environment. Compared with most existing methods, the innovation of the proposed method lies in

(1) The proposed method utilizes GAN for data resampling to address the issue of imprecise detection caused by limited data samples for network attack types to overcome the impact of data imbalance.
(2) Due to the fact that most existing detection methods rarely consider the spatiotemporal characteristics of

* **Corresponding author: Jing Yang,** School of Mathematics and Computer Science, Hanjiang Normal University, Shiyan, 442000, Hubei, P. R. China, e-mail: yangjing@hjnu.edu.cn

traffic, the detection performance in complex environments is poor. Therefore, in the feature extraction stage, the proposed method uses a one-dimensional (1D) multi-scale convolutional layer to extract the spatial features of samples, uses gated recurrent units (GRU) to extract the temporal features of samples, and fuses them to improve the model's representation ability.

(3) To prevent network degradation, the proposed method adds identity mapping to the spatiotemporal residual network, allowing deep networks to integrate features from shallow, middle, and high-level networks, further improving detection reliability.

2 Related work

At present, NID models are studied based on data statistical mining, ML, and DL [8]. The NID model based on data statistical mining captures a large amount of data feature information from network traffic and extracts information and knowledge beneficial to NID through data mining models, such as association analysis, clustering analysis, etc. As proposed by Salah [9], a real-time hardware IDS implemented on FPGA and an algorithm for classifying features from network traffic. Babu et al. [10] proposed an intrusion detection method using clustering algorithms to improve the performance of routing. Jianwu et al. [11] proposed an IDS based on fuzzy theory and an improved apriori algorithm. The fuzzy set technology is used to solve the problem of large boundaries in continuous data segmentation. The above methods are mostly based on statistical methods, and the detection concept is relatively traditional, making it difficult to adapt to complex network environments.

In the context of increasingly complex data, in the late twentieth century, ML algorithms began to be applied in the field of intrusion detection, including Bayesian networks, etc. [12]. Huang et al. [13] proposed an IDS HiDE-IDS based on hierarchical differential evolution to identify unknown network attacks, in which a hierarchical differential evolution algorithm and a new filtering mechanism are designed to identify known and unknown network attacks. Xu et al. [14] proposed an IDS based on XGBoost and Bayesian networks, in which binary grey wolf optimization algorithm and recursive feature elimination are used as target variables to select the most relevant feature subset, XGBoost is used for data classification, and a Bayesian optimized tree structure Parzen estimator is used to optimize model hyperparameters. Jin et al. [15] proposed a new IDS framework based on federated learning, deepening the memory of the overall old class and using knowledge distillation methods to enhance the local model memory of each specific class. Sadhwani et al. [16] proposed a compact and lightweight IDS that combines ExtraTreeClassifier with a new data preprocessing method to achieve network anomaly detection. The above methods make it difficult to handle the characteristics of complex, high-dimensional, and nonlinear data well, resulting in low detection accuracy and being greatly affected by factors such as noise.

With the rapid improvement of computer computing power, DL has achieved many results and has also been applied in the field of NID. According to Sravanthi and Kumar [17], a deep belief network (DBN)-based IDS method is proposed, which combines the cuckoo search algorithm and lion algorithm to optimize the DBN weights to ensure detection reliability. Kumar and Sharma [18] proposed a hybrid intelligent system and a hierarchical network classifier based on an inverted hourglass. In addition, a hybrid optimization feature selection technique only selects features that can improve detection accuracy. Finally, a hierarchical network model based on an inverted hourglass is used for classification, which upsamples the data as the number of layers increases. Odeh and Taleb [19] utilized an integrated DL framework for NID, which includes convolutional neural networks (CNNs), long short-term memory (LSTM) networks, and GRUs. Voting strategies are integrated into the framework to facilitate hierarchical computation and learning. Gu et al. [20] used a deep denoising autoencoder to extract feature representations, and a network was used to achieve multi-type detection of balanced data after dimensionality reduction. Sreekanth et al. [21] proposed an IDS based on federated deep reinforcement learning, where multiple agents are deployed in a distributed manner on the network, and each agent runs a deep Q-network logic. The above methods have achieved good detection results using DL algorithms but lack comprehensive analysis of the spatiotemporal characteristics of data, and the detection effect needs further improvement.

Based on the above analysis, most existing intrusion detection methods have problems such as poor feature extraction and significant impact from data imbalance. Therefore, the proposed method utilizes GAN for data resampling to overcome the impact of data imbalance. Use 1D multi-scale convolutional layers to extract spatial features of data samples and GRU to extract temporal features of data samples and fuse them while adding identity maps to improve the model's representation ability.

3 System model

IDS is an active defense security tool that combines hardware and software. It ensures security by disrupting attackers' access to information or preventing them from further accessing network systems and can effectively resist network attacks. An intrusion data intelligent detection system is proposed for complex network states in the big data environment, and its overall architecture is shown in Figure 1.

Among them, the system obtains real-time operational data of complex networks and inputs it into the logic layer and database. The logic layer is mainly responsible for preprocessing and feature extraction of data, as well as storing the processed data, while connecting to the database. The management is mainly responsible for detecting intrusion data and displaying detection results through front-end pages to respond with defensive measures.

4 Wireless sensor network data preprocessing and resampling

4.1 Data preprocessing

First, use One-Hot encoding to convert the string type features in the dataset into numerical types. One-Hot encoding uses an N-bit state register to expand the encoding of N states, each with its own separate register bit. One-Hot encoding represents binary vectors as classification variables and maps classification values to integer values [22]. The value of discrete features can be extended to Euclidean space.

After numerical processing, check for any empty values. If there are no empty values, perform standard normalization on the data. The proposed method adopts the Z-score normalization method. The standard deviation σ of Z-score is calculated as follows:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}, \tag{1}$$

where x is the input value, N is the total amount of data, and μ is the mean value.

The Z-score standardization conversion is as follows:

$$z = \frac{x - \mu}{\sigma}. \tag{2}$$

4.2 Data resampling based on GAN

Due to the large amount of data traffic in NID, and most of it being non-attack-type data, training models using existing data are not only time-consuming, but also due to data imbalance, there may be overfitting or underfitting problems in the model, which affects the effectiveness of traffic detection. GAN can be used to generate minority class samples, reducing the problem of low detection efficiency caused by data imbalance [23]. GAN consists of two models: a generator and a discriminator. The generator learns the distribution of actual data, and the discriminator determines whether the input results come from actual data or the generator. The two are further improved during the actual training process, and their generation and discrimination abilities are respectively cultivated. The main inspiration for this network comes from the Nash equilibrium of game theory, and the model is trained to achieve the Nash equilibrium state. The data resampling process based on GAN is shown in Figure 2.

For generator module G, LSTM can overcome the problem of gradient vanishing. Therefore, a three-layer LSTM network is chosen as the generator module. Meanwhile, a multi-layer perceptron with two hidden layers is used as discriminator D. The discriminator updates the parameters of the model by reducing the cross entropy between the

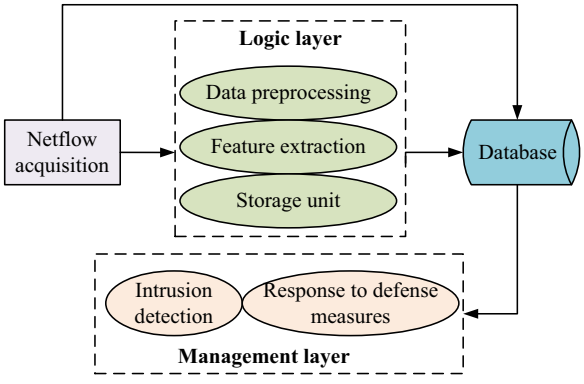

Figure 1: Overall framework of the IDS.

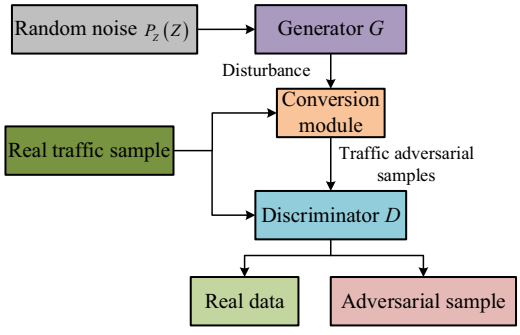

Figure 2: Data resampling process based on GAN.

correct classification of sample x and the estimated distribution $P_{\text{model}}(y|x)$.

In the data resampling process based on GAN, generator G is used to generate samples similar to the original samples, discriminator D is used to distinguish between the generated samples and the original samples, and generator G and discriminator D continuously optimize their own parameters through game confrontation to generate adversarial samples that can be close enough to the true sample distribution. During this process, the generator module G and discriminator module D are alternately optimized. The judgment results of discriminator D can guide the parameter adjustment of generator G. Therefore, in a single round of network training, after the completion of discriminator D training, the parameters of generator G are optimized by freezing the parameters of discriminator D to generate samples. After multiple rounds of training are completed, generator G can generate network traffic adversarial samples that can be distorted as real.

5 Intrusion detection method based on improved spatiotemporal residual network

5.1 Improved spatiotemporal residual network

To learn the spatiotemporal characteristics of network traffic, the proposed method proposes an improved spatiotemporal residual network, which utilizes multi-scale 1D convolutional layers to increase network width and extract spatial features, extracting temporal features between data through GRU and fusing spatial and temporal features to enhance model representation and generalization capabilities [24,25]; by stacking multiple spatiotemporal residual modules to increase network depth, adding identity maps, and fusing shallow, middle, and high-level features of the network, the information hierarchy is enriched, avoiding problems such as vanishing gradients, exploding gradients, and network degradation in deep networks. The structure of the improved spatiotemporal residual network is shown in Figure 3.

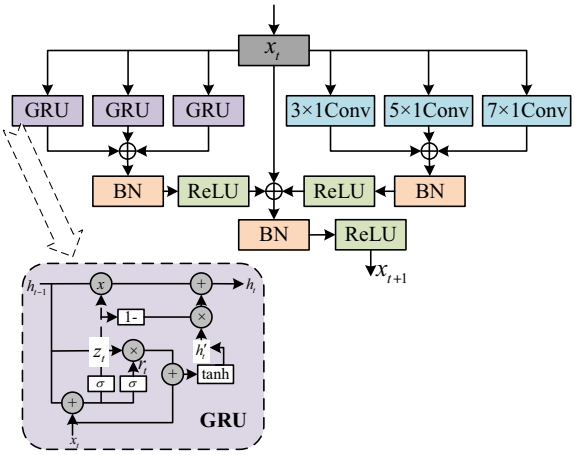

Figure 3: Structure of improved spatiotemporal residual network.

5.1.1 Multi-scale 1D convolutional layer

The detection of network traffic cannot rely solely on discrete local features but should extract traffic features of different sizes through multiple convolution kernels of different scales and fuse them to obtain multiple sets of local features. The proposed method achieves spatial feature extraction of multi-scale network traffic data through multi-scale 1D convolutional layers, using convolutional filters with lengths of 3, 5, and 7, respectively. Multi-scale spatial feature fusion is achieved by adding them item by item, increasing the information content of individual elements while keeping the feature dimension unchanged, reducing subsequent computational costs. Using BN layers for batch normalization, the data follows or approximates a standard normal distribution, accelerating the convergence speed of neural networks and preventing gradient explosion, vanishing gradients, and overfitting phenomena. Choosing ReLU as the activation function amplifies the differences between features, increases network sparsity, and makes the final extracted multi-scale spatial fusion features more representative, improving the network's generalization ability.

5.1.2 GRU

The proposed method uses GRU to extract the dependency relationships between network traffic data, preventing the problems of exploding gradients. GRU has fewer parameters. Divide the data into a window size of W, which includes W consecutive network traffic sample data U, which can be represented as

$$U = [x_{t-W+1}, x_{t-W+2}, \ldots, x_t]. \tag{3}$$

Inputting U into GRU will generate a vectorized representation of the hidden state h_n for each time step data. The output of GRU is as follows:

$$h_1, h_2, \ldots, h_n = \text{GRU}(x_1, x_2, \ldots, x_n). \tag{4}$$

5.1.3 Residual learning

The proposed method achieves residual fitting by constructing a skip connection method of "direct/identity mapping." The improved spatiotemporal residual mapping is as follows:

$$x_{\text{out}} = x + F(x) = x + x_{\Sigma\text{out}}, \tag{5}$$

where x and x_{out} are the inputs and outputs for improving spatiotemporal residual learning, respectively; $x_{\Sigma\text{out}}$ is the spatiotemporal fusion feature extracted by the network; and $F(x)$ represents residual mapping.

Similarly, batch normalization is performed using BN, and ReLU is selected as the activation function to optimize the data distribution and obtain the final output of the improved spatiotemporal residual network. After adding identity mapping, as the network deepens, the network performance does not decrease, and the deep network performs better than the shallow network, solving the problem of network degradation.

5.2 NID process

The overall process of intrusion detection using the proposed method is shown in Figure 4.

(1) Data preprocessing: Perform data processing such as One-Hot encoding and data normalization on the original dataset. After numerical processing, the feature dimension of the NSL-KDD dataset increased to 122 dimensions; the feature dimension of the UNSW-NB15 dataset has been increased to 196 dimensions; the CIC-IDS2017 dataset does not require conversion and remains at 78 dimensions. Finally, each processed network packet contains 122-dimensional, 196-dimensional, and 78-dimensional feature attributes, as well as 1-dimensional type labels.

(2) Data resampling: Using a GAN network to generate new attack samples of a specified type based on the category labels of the attack samples, the newly obtained attack data is fused with the original training set to obtain a new balanced training set.

(3) Model detection: Using multi-scale 1D convolutional layers to extract spatial features, and using GRU to extract temporal features between data, and fusing spatial and temporal features. By stacking multiple spatiotemporal residual modules to increase network depth, identity mapping is added to fuse shallow, middle, and high-level features of the network. Afterwards, a softmax classifier is used for classification, and the abnormal traffic detection results are obtained by training the cross entropy loss function.

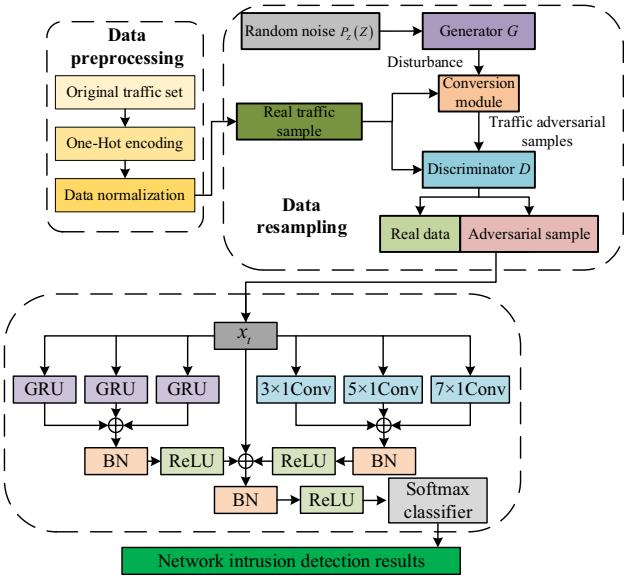

Figure 4: NID process based on improved spatiotemporal residual network and GAN.

6 Experiment and analysis

6.1 Experimental environment

The experimental environment is the Win10 operating system, with a 4-core 8-thread Intel (R) Core (TM) i7-10510U CPU @ 1.80 GHz processor, 12 GB of memory, and Python 3.6 as programming language. Tensorflow GPU-1.15.0 is used in the experiment to build the model. To ensure the training efficiency of the model, the time step is set to 4.

6.2 Experimental dataset

Three datasets are selected for evaluation in the experiment, namely the NSL-KDD, UNSW-NB15, and CIC-ISDS2017 datasets. The features and attack identifiers of NSL-KDD and KDD99 are the same, but NSL-KDD has cleared and organized some duplicate records in KDD99, containing a total of over 100,000 pieces of data. The data distribution is shown in Table 1.

UNSW-NB15 is a dataset collected in 2015 by the Canberra Network Range Laboratory, where the traffic data is more in line with current real network activity and contemporary

Table 1: Sample distribution of the NSL-KDD dataset

Attack type	Data volume
Normal	77,054
DoS	53,471
Probe	14,077
R2L	3,749
U2R	252
Total	148,517

Table 2: Sample distribution of the UNSW-NB15 dataset

Attack type	Data volume
Normal	93,000
Generic	58,871
Exploits	44,525
Fuzzers	24,246
DoS	16,353
Reconnaissance	13,987
Analysis	2,677
Backdoor	2,329
Schellcode	1,511
Worms	174
Total	257,673

attack behavior. This dataset contains 9 types of attacks, 49 features, and 1 labeled feature. Table 2 shows the data distribution after dividing the dataset into training and testing sets in a 3:2 ratio.

The CIC-IDS2017 dataset is sourced from the Canadian Institute of Cybersecurity's collection of network data from July 3 to 7, 2017, which includes benign and the latest common attacks, filling the gap in the UNSW-NB15 dataset where there is no network-based attack. There are a total of 15 types in the CIC-IDS2017 dataset, and the ones with similar abnormal attack properties are merged. For example, the three types of network attacks in the dataset are merged into network attack types. The final dataset has a total of nine traffic types. The data distribution of the CICIDS2017 dataset is shown in Table 3.

6.3 Evaluating indicator

The evaluation indicators used in the experiment include accuracy (Acc), precision (Pre), recall (Rec), and $F1$ score, which are calculated as follows:

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{6}$$

Table 3: Sample distribution of the CICIDS2017 dataset

Attack type	Data volume
BE-NIGN	80,000
DoS	10,787
Portscan	8,000
DDoS	6,267
Patator	13,835
Bot	1,966
Web attack	2,180
Infiltration	36
Heartbleed	19
Total	123,090

$$\text{Pre} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{7}$$

$$\text{Rec} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{8}$$

$$F1 = \frac{2 \times \text{Pre} \times \text{Rec}}{\text{Pre} + \text{Rec}}, \tag{9}$$

where TP is the true example, TN is the true negative example, FP is the false positive example, and FN is the false negative example.

At the same time, algorithm complexity is included in the evaluation criteria, which includes time complexity and space complexity. Time complexity refers to the time consumed by algorithm execution, while space complexity refers to the memory space required by the algorithm.

6.4 Model training

6.4.1 Analysis of the impact of different hyperparameters

The loss value of the proposed method varies depending on the number of iterations. Taking the NSL-KDD dataset as an example, its correlation curve is shown in Figure 5.

As shown in Figure 5, as the number of iterations increases, the number of updates to the weights of the improved spatiotemporal residual network also increases. The network gradually enters an optimized fitting state from underfitting, and during this process, the loss value of the validation set shows a spiral downward trend. When the number of iterations is 11, the loss value is minimized, and the network training reaches the optimal state. When the number of iterations exceeds 11, overfitting occurs, the model loss value begins to increase, the detection performance decreases, and the training time also increases. Therefore, set the number of iterations to 11.

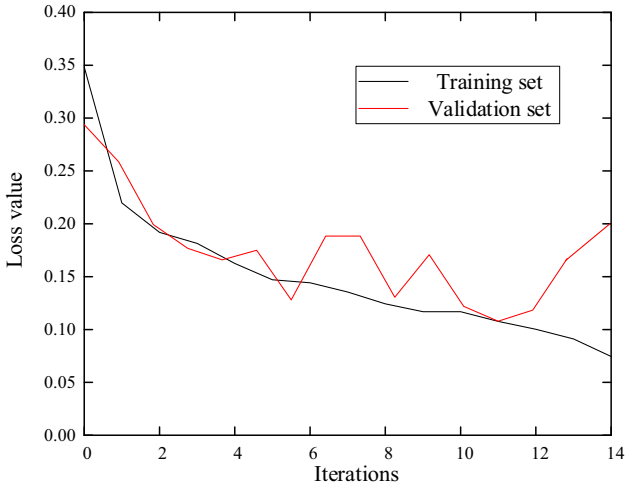

Figure 5: Influence curve of iterations on loss values.

6.4.2 Analysis of the impact of different learning rates

The Adam optimizer can adjust the learning rate on its own, but the initial learning rate still needs to be set through experiments. In the experiment, three learning rates of 0.01, 0.001, and 0.0001 were used for testing. Taking the UNSW-NB15 dataset as an example, the detection loss values are shown in Figure 6.

As shown in Figure 6, when the learning rate is 0.01, the fitting effect of the model is poor, and the loss value fluctuates significantly with high values. When the learning rate is 0.0001, the model loss value stabilizes at around 0.27, indicating poor performance. When the learning rate is 0.001, the model achieves ideal detection performance, with a steady decrease in loss rate and finally stabilizing at around 0.11. Therefore, the initial learning rate is set to 0.001.

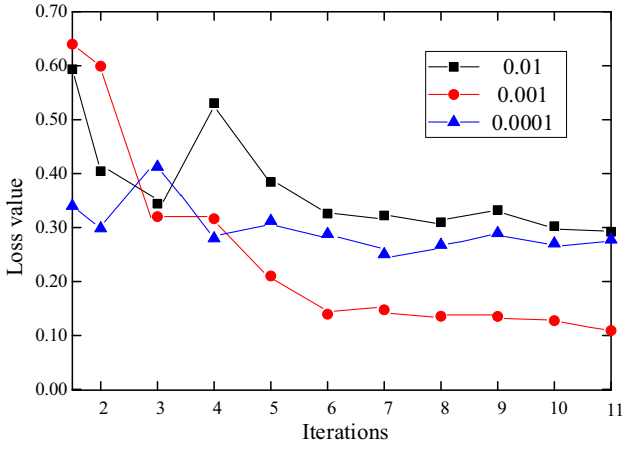

Figure 6: Changes in loss values under different learning rates.

Table 4: Different flow ratios before and after resampling

Comparison	Attack type	Number of attacks	Ratio (%)
Before resampling	BE-NIGN	60,000	69.74
	DoS	8,035	9.34
	Portscan	6,000	6.97
	Patator	12,000	13.95
After resampling	BE-NIGN	60,000	48.40
	DoS	17,656	14.24
	Portscan	22,061	17.80
	Patator	24,247	19.56

6.5 Comparison of data resampling effects

The proportion of attack traffic in the data file is too small, resulting in an imbalance in the proportion. Therefore, taking the CICIDS2017 dataset as an example, four types of traffic with a large amount of data (BE NIGN, Patator, Portscan, and DoS) are selected as the experimental subjects in the experiment. Use GAN to resample the original imbalanced data and obtain a new dataset. The different traffic proportions before and after resampling are shown in Table 4.

According to Table 4, after data resampling by GAN, the proportion of data is uniform, with Patator, Portscan, and DoS attack types all accounting for about 15%, significantly reducing the difference in proportion compared to BE-NIGN. Meanwhile, in order to verify the impact of data resampling on detection performance, tests were conducted on three datasets, and the intrusion detection results before and after resampling are shown in Table 5.

According to Table 5, the performance of the detection method using GAN for resampling has been greatly improved. The detection accuracy of the NSL-KDD, UNSW-NB15, and CICISDS2017 datasets all reached 99.62, 83.98, and 99.86%, which are 5.44, 4.53, and 5.23% higher than before resampling, respectively. This is because after resampling, the dataset

Table 5: Comparison of detection results before and after resampling

Comparison	Index	NSL-KDD	UNSW-NB15	CIC-IDS2017
Before resampling	Acc (%)	94.18	79.45	94.63
	Pre (%)	93.75	79.08	94.37
	Rec (%)	94.32	79.61	94.99
	*F*1 (%)	94.03	79.34	94.68
After resampling	Acc (%)	99.62	83.98	99.86
	Pre (%)	99.43	83.62	99.57
	Rec (%)	99.94	84.15	99.91
	*F*1 (%)	99.68	83.88	99.74

data is more balanced, and the detection model will not have overfitting problems, ensuring detection reliability.

6.6 Comparison and analysis with other methods

To demonstrate the detection performance of the proposed method, it is compared with the methods in the literature [14,17, 19], where the comparison methods are abbreviated as BXGB, IDBN, and CNN-LSTM-GRU.

6.6.1 Comparison of NSL-KDD dataset results

The detection accuracy of four methods on various attack types in the NSL-KDD dataset is shown in Figure 7.

As shown in Figure 7, the proposed method has good detection results for various types of attacks in the NSL-KDD dataset, with an accuracy rate of over 95%. This is because the proposed method uses an improved GAN for data augmentation and utilizes an improved spatiotemporal residual network to deeply analyze the spatiotemporal characteristics of the data. The CNN-LSTM-GRU model integrates CNN, LSTM, and GRU networks for intrusion detection and has good detection performance. However, it does not consider the impact of data imbalance, so the detection performance for R2L and U2R is poor, with an accuracy of less than 95%. The detection performance of IDBN and BXGB is similar, but due to the relatively simple detection model, it is difficult to accurately process network data in big data environments, resulting in relatively low detection accuracy.

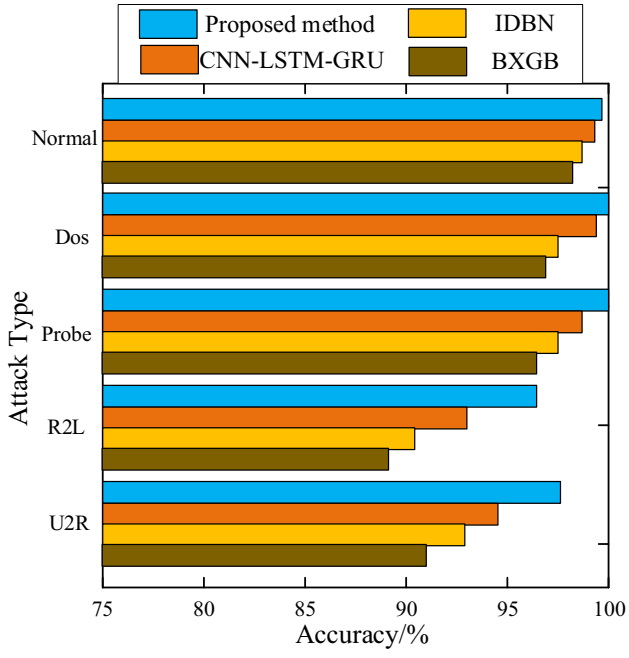

Figure 7: Detection results of the NSL-KDD dataset.

After multiple experiments, the detection results of the four methods on the NSL-KDD dataset are shown in Table 6.

According to Table 6, the detection results of the proposed method are superior to other methods, with accuracy, precision, recall, and *F*1 values of 99.62, 99.43, 99.94, and 99.68%, respectively. CNN-LSTM-GRU has achieved accurate intrusion detection but lacks consideration for imbalanced data, resulting in a 3.54% decrease in detection accuracy compared to the proposed method. Additionally, the model is complex, resulting in a longer detection time, reaching 14.72 s. The improved DBN model in IDBN is easy to train, so the detection time is the shortest, only 8.59 s, but the detection effect is slightly reduced. BXGB combines XGBoost and Bayesian networks for intrusion detection. The training time of the fusion model is 1.68 s longer than that of IDBN, and the model makes it difficult to extract high-dimensional data features, which affects the detection results. Its accuracy is only 92.82%.

6.6.2 Comparison of UNSW-NB15 results

The detection accuracy of four methods on various attack types in the UNSW-NB15 is shown in Figure 8.

From Figure 8, the proposed method has higher detection accuracy for various types of attacks in the UNSW-NB15 than other comparison methods, especially for attack types such as Analysis and Schellcode with fewer samples, and its detection advantage is more obvious. Due to the fact that the UNSW-NB15 records the real network environment, with more types of attacks and more complex features, the detection efficiency of all four methods has

Table 6: Comparison of results between different methods on the NSL-KDD dataset

Method	BXGB	IDBN	CNN-LSTM-GRU	Proposed method
Accuracy (%)	93.37	94.14	96.08	99.62
Precision (%)	92.82	93.61	95.56	99.43
Recall (%)	93.48	94.35	96.11	99.94
*F*1 (%)	93.15	93.98	95.83	99.68
Detection time (s)	10.27	8.59	14.72	10.13
Occupied memory (M)	321	196	215	304

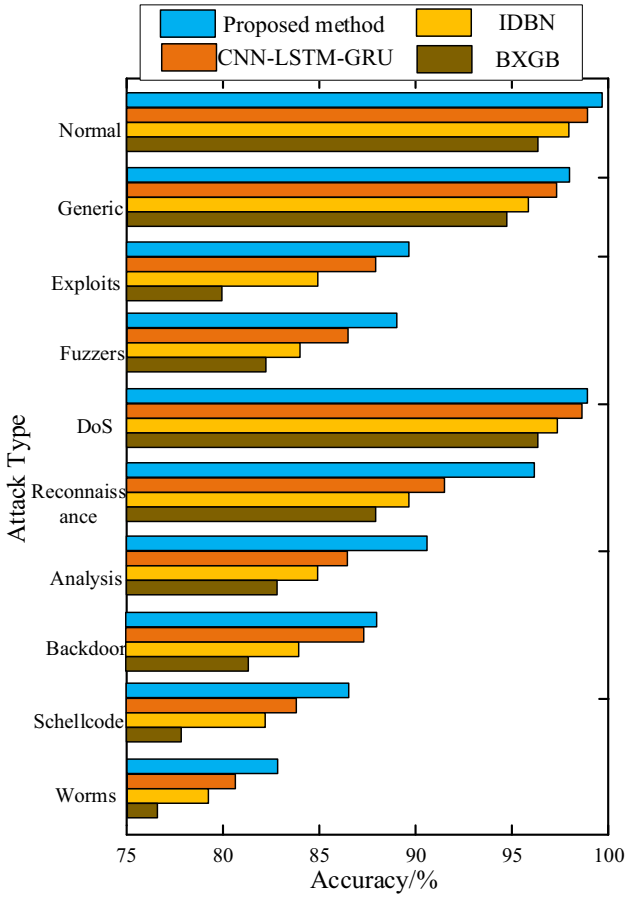

Figure 8: Detection results of the UNSW-NB15.

decreased. After multiple experiments, the detection results of the four methods on the UNSW-NB15 are shown in Table 7.

As shown in Table 6, compared to the NSL-KDD, the proposed method showed a significant decrease in detection performance on the UNSW-NB15 dataset, but still outperforms other comparison methods. Its Acc, Pre, Rec, and *F*1 values are 83.89, 83.62, 84.15, and 83.88%, respectively,

Table 7: Comparison of results between different methods on the UNSW-NB15

Method	BXGB	IDBN	CNN-LSTM-GRU	Proposed method
Accuracy (%)	75.34	76.29	80.72	83.98
Precision (%)	74.98	75.81	80.45	83.62
Recall (%)	75.77	76.63	80.91	84.15
*F*1 (%)	75.37	76.22	80.68	83.88
Detection time (s)	17.43	14.69	20.84	16.71
Occupied memory (M)	601	327	792	563

and the detection time is 16.71 s. Due to the insufficient learning depth of the detection models, BXGB and IDBN have limited network analysis capabilities in big data environments, resulting in a significant decrease in detection performance, with all indicators below 80%. CNN-LSTM-GRU adopts an integrated DL model for intrusion detection, which can ensure detection reliability. However, due to data imbalance, its *F*1 value decreases by 3.20% compared to the proposed method, and the detection efficiency is also low.

6.6.3 Comparison of CIC-IDS2017 results

The detection accuracy of four methods on various types of attacks in the CIC-IDS2017 is shown in Figure 9.

From Figure 9, due to the merging of anomalies with similar attack properties in the CIC-IDS2017 dataset, the detection performance of the detection method has been significantly improved compared to the UNSW-NB15 dataset. The proposed method has a detection accuracy of nearly 100% for most attack types, and the effect is ideal. For Infiltration and Heartbleed small sample data, the detection

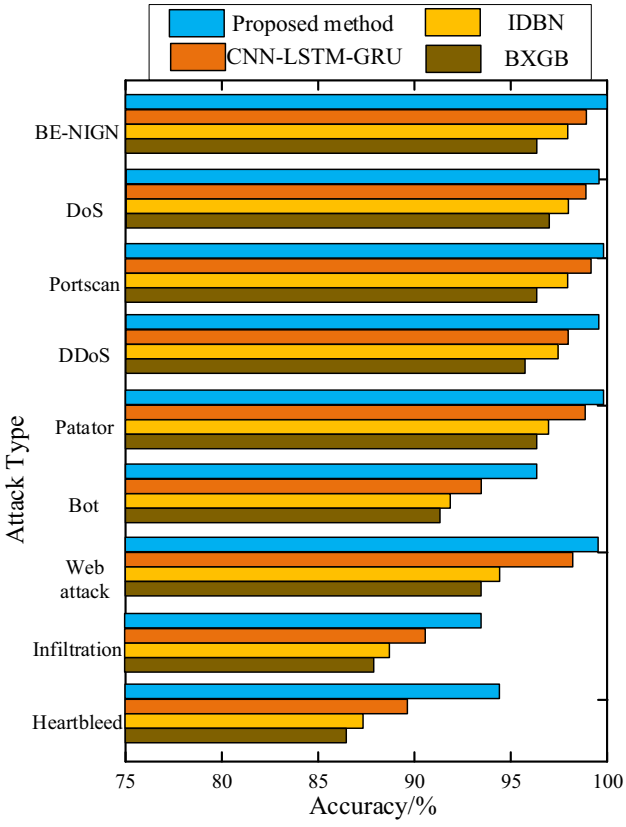

Figure 9: Detection results of the CIC-IDS2017.

Table 8: Comparison of results between different methods on the CIC-IDS2017

Method	BXGB	IDBN	CNN-LSTM-GRU	Proposed method
Accuracy (%)	93.79	94.51	96.43	99.86
Precision (%)	93.36	94.22	96.18	99.57
Recall (%)	94.18	95.06	96.95	99.91
*F*1 (%)	93.77	95.14	96.56	99.74
Detection time (s)	12.91	10.37	16.19	12.45
Occupied memory (M)	426	251	524	391

accuracy of the other three comparison methods is less than 90%. After multiple experiments, the detection results of the four methods on the CIC-IDS2017 are shown in Table 8.

As shown in Table 8, the four methods achieved the best detection performance on the CIC-ISDS2017, because the dataset was close to the real situation and the attack type was set reasonably. The Acc, Pre, Rec, *F*1 value, and detection time of the proposed method are 99.86%, 99.57%, 99.91%, 99.74%, and 12.45 s, respectively. The proposed method fully considers the spatiotemporal characteristics of network data and utilizes the GAN model to solve the problem of data imbalance. Therefore, the detection effect is the most ideal. Taking the *F*1 value as an example, it improves by 5.97, 4.60, and 3.18% compared to BXGB, IDBN, and CNN-LSTM-GRU, respectively. However, due to the complexity of the proposed model, it occupies a large memory space, reaching 391 M. Overall, the proposed method is effective for intrusion detection in complex networks in big data environments.

6.7 Ablation experiment

Evaluate the contribution of each module of the proposed method through ablation experiments. Taking the UNSW-NB15 dataset as an example, the results of the ablation experiments are shown in Figure 10.

According to Table 9, the introduction of multi-scale 1D convolution and GRU for spatiotemporal feature extraction in the residual network has greatly improved the detection accuracy by 3.73%. At the same time, by adding identity mapping to improve the spatiotemporal residual network, the problem of network degradation is solved, with a detection accuracy of 79.45%. The fusion of GAN effectively solves the problem of data imbalance, so the detection accuracy of the proposed method reached 83.98%. The proposed method deeply combines the spatiotemporal characteristics of network traffic

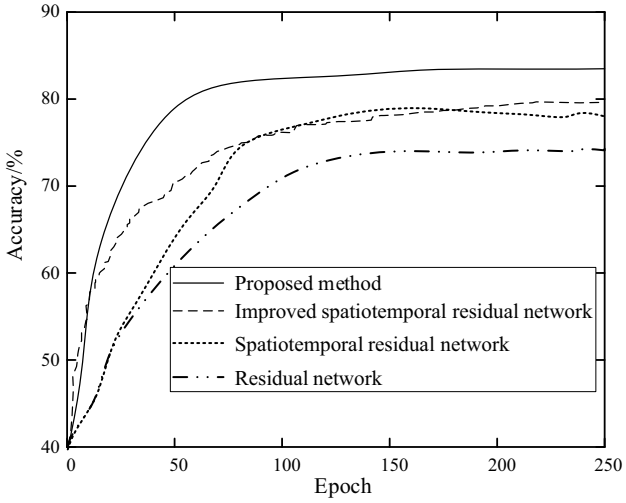

Figure 10: Results of ablation experiment.

and greatly reduces the impact of data imbalance, making the NID performance ideal.

6.8 Analysis of application results in different scenarios

To demonstrate the scalability of the proposed method, three scenarios of power supply companies, hospitals, and Internet companies were selected for testing, and the network data of three sites for 1 week were selected as test sets, respectively. The intrusion detection accuracy of the four methods is shown in Table 9.

According to Table 9, the proposed method can maintain the best intrusion detection accuracy in different scenarios. Due to the use of a dedicated power grid by power companies, which has a certain ability to resist network attacks, the detection accuracy of the proposed method is as high as 99.54%. Internet companies are vulnerable to network attacks from all walks of life, and the environment is complex, so the detection accuracy rate has

Table 9: Intrusion detection results in different scenarios

Scenarios	Accuracy (%)		
	Power supply company	Hospital	Internet company
BXGB	91.17	88.75	80.92
IDBN	94.09	91.91	84.87
CNN-LSTM-GRU	97.65	94.38	88.15
Proposed method	99.54	96.82	92.39

declined. The detection accuracy rate of the BXGB model is only 80.92%.

7 Conclusion

At present, artificial intelligence technologies such as DL have shown good performance in intrusion detection. However, due to the imbalanced categories and high noise content in traffic data, the accuracy of intrusion detection methods is seriously affected. Therefore, an NID method that integrates improved spatiotemporal residual networks and GANs in a big data environment is proposed. Among them, in the big data environment, GAN is used for data resampling to generate a balanced sample set, which is input into an improved spatiotemporal residual network model for analysis, thereby obtaining the intrusion detection results of the network. The experimental results based on the NSL-KDD, UNSW-NB15, and CICIDS2017 datasets show that

(1) The GAN model has alleviated the problem of data imbalance, and the detection performance has been improved after data resampling. The detection accuracy of the three datasets has all reached 99.62, 83.98, and 99.86%, which are 5.44, 4.53, and 5.23% higher than before resampling, respectively.
(2) The improved spatiotemporal residual network further improves detection performance by fully extracting the spatiotemporal characteristics of the data and adding identity maps. Taking the CIC-ISDS2017 dataset as an example, the proposed method has an Acc, Pre, Rec, $F1$ value, and detection time of 99.86%, 99.57%, 99.91%, 99.74%, and 12.45 s, respectively. Overall performance is superior to other comparative methods.

Due to the high computational complexity and long detection time of the proposed method, in the subsequent work, the model structure will be further optimized to reduce complexity, and the proposed method will be applied to other datasets for testing to enhance its scalability. In addition, the algorithm can also be applied to specific industries or fields, such as finance, to predict network attacks on systems in these fields, reduce the incidence of network attacks, and thus reduce economic losses in these fields.

Funding information: This work was supported by 2022 Science Research Program Guidance Project "Research and Application of Smart Campus Security Strategy Based on Big Data Technology" (No.B2022217), Hubei Provincial Department of Education and 2022 Provincial Teaching Research Project "Reform and Innovation of the Talent Training Model for Computer Majors in Local Applied Undergraduate Universities under the Background of New Engineering" (No.2022462), Hubei Provincial Higher Education Institute.

Author contribution: The author confirms the sole responsibility for the conception of the study, presented results, and manuscript preparation.

Conflict of interest: The author declares that there is no conflict of interest regarding the publication of this paper.

Data availability statement: The data used to support the findings of this study are included within the article.

References

[1] Z. T. Chen, X. D. Yang, B. Jin, M. Y. Guo, and M. M. Li, "Industrial internet security evaluation technology based on digital twin," *J. Comput. Methods Sci. Eng.*, vol. 22, no. 6, pp. 1981–1994, 2022.
[2] B. Xu, "Design of intrusion detection system for intelligent mobile network teaching," *Comput. Electr. Eng.*, vol. 112, p. 109013, 2023.
[3] S. F. Li, C. Yue, S. H. Liu, Y. P. Lai, Y. D. Zhu, and A. Naveed, "HDA-IDS: A hybrid DoS attacks intrusion detection system for IoT by using semi-supervised CL-GAN," *Expert. Syst. Appl.*, vol. 238, p. 122198, 2024.
[4] L. J. Dong, J. Wang, J. C. Wang, and H. W. Wang, "Safe and intelligent mining: Some explorations and challenges in the era of big data," *J. Cent. South. Univ.*, vol. 30, no. 6, pp. 1900–1914, 2023.
[5] L. L. Guo, L. M. Wang, X. M. Han, L. Yue, Y. H. Zhang, and M. H. Gao, "ROCM: A rolling iteration clustering model via extracting data features," *Neural Process. Lett.*, vol. 55, no. 4, pp. 3899–3922, 2022.
[6] G. Vembu and D. Ramasamy, "Optimized deep learning-based intrusion detection for wireless sensor networks," *Int. J. Commun. Syst.*, vol. 36, no. 13, p. 1, 2022.
[7] G. Ali and F. S. Mostafa, "A deep learning approach to network intrusion detection using a proposed supervised sparse auto-encoder and SVM," *Iran. J. Sci. Technol. Trans. Electr. Eng.*, vol. 46, no. 3, pp. 829–846, 2022.
[8] R. Alkanhel, "Network intrusion detection based on feature selection and hybrid metaheuristic optimization," *Comput. Mater. Continua*, vol. 74, no. 2, pp. 2677–2693, 2022.
[9] S. T. Salah, "A real-time hardware intrusion detection system and a classifying features algorithm," *J. Appl. Secur. Res.*, vol. 18, no. 4, pp. 845–879, 2023.
[10] A. V. S. Babu, P. M. Devi, B. Sharmila, and D. Suganya, "Performance analysis on cluster-based intrusion detection techniques for energy efficient and secured data communication in MANET," *Int. J. Inf. Syst. Change Manag.*, vol. 11, no. 1, pp. 56–69, 2019.
[11] Z. Jianwu, H. Jiasen, and Z. Di, "Intrusion detection model based on fuzzy theory and association rules," *Telecommun. Sci.*, vol. 41, no. 1, pp. 130–139, 2019.
[12] Y. A. Sawafi, A. Touzene, and R. Hedjam, "Hybrid deep learning-based intrusion detection system for RPL IoT networks," *J. Sens. Actuator Netw.*, vol. 12, no. 2, p. 21, 2023.

[13] H. Y. Huang, L. Tao, D. Yong, B. B. Li, and A. Liu, "An artificial immunity based intrusion detection system for unknown cyberattacks," *Appl. Soft Comput.*, vol. 148, p. 110875, 2023.
[14] B. Y. Xu, L. Sun, X. Q. Mao, R. Y. Ding, and C. W. Liu, "IoT intrusion detection system based on machine learning," *Electronics*, vol. 12, no. 20, p. 4289, 2023.
[15] Z. G. Jin, J. Y. Zhou, B. Li, X. D. Wu, and C. X. Duan, "FL-IIDS: A novel federated learning-based incremental intrusion detection system," *Future Gener. Comput. Syst.*, vol. 151, pp. 57–70, 2024.
[16] S. Sadhwani, B. Manibalan, and R. Muthalagu, "A lightweight model for DDoS attack detection using machine learning techniques," *Appl. Sci.*, vol. 13, no. 17, p. 9937, 2023.
[17] G. Sravanthi and M. S. Kumar, "A weight optimized deep learning model for cluster based intrusion detection system," *Opt. Quantum Electron.*, vol. 55, no. 14, p. 1224, 2023.
[18] N. Kumar and S. Sharma, "A hybrid modified deep learning architecture for intrusion detection system with optimal feature selection," *Electronics*, vol. 12, no. 19, p. 4050, 2023.
[19] A. Odeh and A. A. Taleb, "Ensemble-based deep learning models for enhancing IoT intrusion detection," *Appl. Sci.*, vol. 13, no. 21, p. 11985, 2023.
[20] Y. H. Gu, Y. Yang, Y. Yan, F. Shen, and M. Gao, "Learning-based intrusion detection for high-dimensional imbalanced traffic," *Comput. Commun.*, vol. 212, pp. 366–376, 2023.
[21] V. Sreekanth, S. Kamalakanta, and M. Dinesh, "Federated reinforcement learning based intrusion detection system using dynamic attention mechanism," *J. Inf. Secur. Appl.*, vol. 78, p. 103608, 2023.
[22] M. Jawad, L. Ming, and R. Mudassar, "An accurate detection of tool wear type in drilling process by applying PCA and one-hot encoding to SSA-BLSTM model," *Int. J. Adv. Manuf. Technol.*, vol. 118, no. 11–12, pp. 3897–3916, 2021.
[23] Z. Ting, G. S. Hu, Y. Yi, and D. Yi, "A super-resolution reconstruction method for shale based on generative adversarial network," *Transp. Porous Media*, vol. 150, no. 2, pp. 383–426, 2023.
[24] S. Balaji and S. Sankaranarayanan, "Hybrid distributed deep-GAN intrusion detection system in IoT with Autoencoder," *Int. J. Fuzzy Syst. Appl. (IJFSA)*, vol. 11, no. 4, pp. 1–20, 2022.
[25] X. X. Liao, J. Y. Yuan, Z. M. Cai, and J. H. Lai, "An attention-based bidirectional GRU network for temporal action proposals generation," *J. Supercomput.*, vol. 79, no. 8, pp. 8322–8339, 2022.