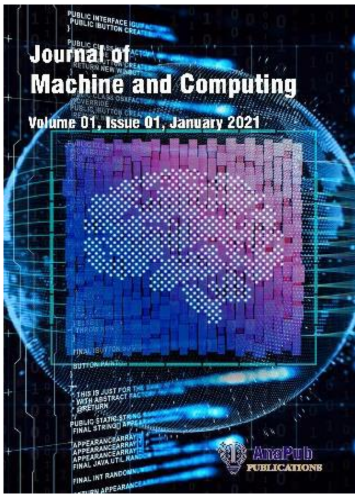
Journal Pre-proof



Research on Improved LSTM and Deep Learning Intrusion Detection Algorithms

Baoguo Liu, Eric B. Blancaflor and Mideth Abisado

DOI: 10.53759/7669/jmc202505006

Reference: JMC202505006

Journal: Journal of Machine and Computing.

Received 10 May 2024

Revised form 23 August 2024

Accepted 30 September 2024

Please cite this article as: Baoguo Liu, Eric B. Blancaflor and Mideth Abisado, “Research on Improved LSTM and Deep Learning Intrusion Detection Algorithms”, Journal of Machine and Computing. (2025). Doi: https://doi.org/10.53759/7669/jmc202505006

AnaPub
PUBLICATIONS

Research on Improved LSTM and Deep Learning Intrusion Detection Algorithms

Baoguo Liu[1],*, Eric B. Blancaflor [2], Mideth Abisado [3]

[1] National University, Philippines, liub@students.national-u.edu.ph
[2] Mapua University, Philippines, ebblancaftor@national-u.edu.ph
[3] National University, Philippines, mbabisado@national-u.edu.ph
*Corresponding author

Abstract:
Purpose: These days, network security concerns are becoming more and more important due to the Internet's quick development. The goal of this article is to enhance the feature extraction and classification accuracy of network intrusion detection models by addressing the issues of low classification accuracy and weak generalization ability of current models in the field. Methods: A deep learning network intrusion detection model and an LSTM model based on convolutional neural networks (CNN) and weight dropout, abbreviated as AWD-CNN-LSTM, are creatively proposed. This model effectively extracts nonlinear features from the dataset using CNN, and temporal features from the dataset using LSTM. To alleviate overfitting caused by data imbalance, GP-GAN is introduced to oversample rare types of data, further enhancing the model's generalization ability. The proposed intrusion detection model was experimentally tested on the NSL-KDD dataset.
Results: The experimental results showed that the proposed method has better accuracy compared to traditional machine learning methods such as SVM and K-Means, as well as deep learning methods such as convolutional neural networks, regardless of whether it is related to random forests.
Conclusion: The improved accuracy and F1 score performance suggest that the IDS model suggested in this article has some practical value and can be used to enhance network security protection capabilities through network intrusion detection.

Keywords: Network Security, AWD-CNN-LSTM, GP-GAN, Deep Learning, Intrusion Detection Systems

Introduction

The significance of network security has grown with the quick development of cutting-edge technologies like network technology and digitization. How to accurately identify various network attack behaviors has become a research hotspot in the field of network security. Existing firewalls Network security devices such as WAF are no longer able to meet the increasingly severe demands of network security situations, and intrusion detection technology is becoming increasingly widely used and its role is becoming more and more important. At present, intrusion detection technology is constantly evolving based on various technological means (such as deep learning), and it is necessary to develop better defense measures or algorithms for different network attacks to cope with current network attacks.

Intrusion detection system(IDS) can create an effective defense barrier in the network system, which can be used to automatically detect and classify intrusion, attack, or violation of security policy in time on the Internet and its internal network. IDS can autonomously monitor and extract features from network traffic data, effectively detecting hidden attack and unauthorized behavior in network traffic data. It is a tool with active defense technology in the field of network security. Denning[1], The IDS model was first established by others in 1986, which can preliminarily identify abnormal behavior in network environment systems. In addition, traditional machine learning algorithms are widely used in IDS, Serinelli[2], the team established a Support Vector Machine (SVM) model and used it to classify zombie program viruses. Elbasiony[3], A hybrid network intrusion detection model combining random forest and weighted K-Means was proposed by others, which effectively identifies and alerts illegal intrusions in network systems. Shubair [4]el, a flow-based intrusion detection algorithm was proposed by others, which uses the least squares method to reduce errors and uses the KNN algorithm to select the best matching class, achieving good results. However, with the development of deep learning technology, the scale of Internet traffic continues to grow, network data and traffic become multidimensional, and network intrusion becomes more complex, which makes the network IDS applied in the field of machine learning appear in the complex network environment.

Unlike shallow machine learning algorithms, deep learning techniques can effectively handle the complex relationships between high-dimensional network traffic data, without the need for human intervention, and can uncover the inherent connections between data, especially in data dimensionality reduction and classification tasks, with significant advantages. Anlin et al[5], the fusion of convolutional neural network (CNN) and gated recurrent unit (GRU) are applied in intrusion detection models to fuse network data features and extract temporal information features, effectively distinguishing network attack types. Chouikha et al[6] established an intrusion detection model based on bidirectional LSTM, which can extract temporal and spatial features from data. The extracted forward and reverse sequence features are fused, and then the fused features are distinguished and classified. After comparative verification, the classification performance of network data labels is significantly improved. Tao[7] compared to other machine learning models, time convolutional neural networks (TCN) have higher accuracy and lower false alarm rates in extracting temporal information and learning text sequence features when applied to the task of detecting time series information.

The above is the current research status in the field of network intrusion detection. It can be seen that the use of deep learning technology in network intrusion detection systems is far more effective than traditional machine learning algorithms[8]. Deep learning technology has many advantages, but there are still many problems in deep learning research[9], such as:

(1) the existing IDS model has a single feature extraction, and CNN-based network intrusion detection systems have significant effects[10]. However, this model cannot mine the long-distance relationship between features, The network attacks in IDS require high requirements for temporal and spatial feature mining, so it is necessary to consider the long-range or spatial dependency relationship between features.

(2) Data sparsity. In IDS, high-dimensional data is used, and certain features in the data may affect the final experimental results. Extracting the best data features from high-dimensional data directly determines the quality of the IDS model.

(3) Data imbalance, which means that the traffic data of malicious attacks is usually much lower than normal traffic data, and models trained with this data will lead to severe bias and fail to achieve good prediction results.

In response to the problem of data sparsity mentioned above, this paper uses the Principal Component Analysis (PCA) algorithm to extract the main features of the data while reducing the data dimension, to improve the training speed and prediction accuracy of the model. For the problem of single-feature extraction in existing IDS models, this paper creatively proposes the AWD-CNN-LSTM model, which combines the advantages of CNN and LSTM networks and has a significant effect in extracting long-term dependencies between spatial features and long-range temporal features. In response to the problem of data imbalance, this article uses an adversarial network based on a gradient penalty algorithm (GP-GAN) to help maintain a balance between abnormal traffic and real traffic, thus solving the gradient problem in IDS model training.

2. Related Work

2.1 GP-GAN Model

Strong machine learning models called Generative Adversarial Networks (GANs) are frequently used to produce fictitious and fake data to balance abnormal and real data.[11], GAN transforms generative modeling into a game between two networks, and its network structure is shown in Figure 1.

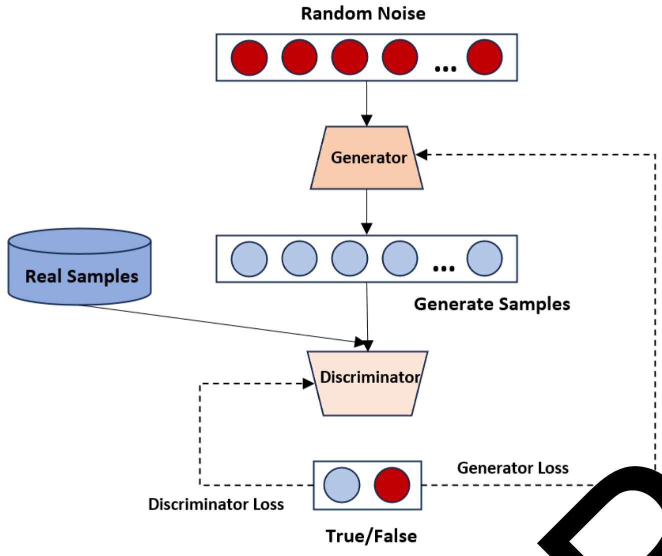


Figure 1 GP-GAN model framework diagram

Among them, Generator(G) uses random noise output to generate samples, while Discriminator (D) mainly distinguishes between real samples and generated samples[12]. The objective function of GAN is based on the mutual game between the generator and discriminator, and the formula for the objective function is as follows:

$$\min_{G}\max_{D} V(D,G) = E_{s\sim P_r}[\log D(x)] + E_{\tilde{s}\sim P_g}[\log(1 - D(\tilde{x}))]$$

In formula (1), $G(z)$ maps the noisy data vector to the data space of the generated samples, $\tilde{s} = G(z)$. P_r represents the true distribution of data, P_g represents the distribution of generated data, $z \sim P_z$ represents the distribution of random noise or Gaussian noise. We assume that the probability of sample s being a true sample is represented by $D(s)$. The function of a discriminator is to distinguish between generated samples and real samples. It will try to increase the value of $D(s)$ and reduce the value of $D(\tilde{s})$ as much as possible, so that the objective function reaches the global optimal solution, which satisfies the condition of $P_g = P_r$. Due to the use of adversarial training methods in GAN, the discriminative results of the discriminator directly affect the training effectiveness of the generator, resulting in unstable training and vanishing gradients in GAN.

Based on the above issues, this article uses the Adversarial Network with the Gradient Penalty Algorithm (GP-GAN), GP-GAN satisfies the Lipschitz continuity condition by adding a gradient penalty term to the loss function[13]. After satisfying the LC condition, it can effectively constrain the rate of change of the function, ensuring that it does not grow indefinitely. This can effectively improve the training stability of GAN and ensure the quality of the generated data. The following formulas 2 and 3 represent the loss functions of the generator and discriminator, respectively:

$$L_G = -E_{\tilde{s}\sim P_g}[D(\tilde{s})] \quad (1)$$

$$L_D = E_{\tilde{s}\sim P_g}[D(\tilde{s})] - E_{\tilde{s}\sim P_g}[D(\tilde{s})] + \lambda E_{\hat{s}\sim P_{\hat{s}}}[(\|\nabla_{\hat{s}} D(\hat{s})\|_2 - 1)^2](2)$$

In the formula, $P_{\hat{s}}$ represents uniform sampling along a straight line between the two points of the real sample and the generated sample, which is represented by the following formula:

$$\hat{s} = \varepsilon s + (1-\varepsilon)\tilde{s} \quad s \in P_r\ ,\ \tilde{s} \in P_g\ ,\ \varepsilon \in U[0,1](3)$$

GAN first inputs random noise into the generator to obtain the generated samples, then passes the generated samples and real samples to the discriminator to obtain the recognition results[14]. Finally, using formulas 2 and 3 to calculate the loss value and perform backpropagation, iterating continuously until the model converges, obtaining high-quality data.

2.2 PCA Algorithm

In response to the problems of data redundancy and feature extraction in IDS, this article adopts principal component analysis (PCA) algorithm. PCA is a commonly used dimensionality reduction method in machine learning, widely used in data analysis and feature extraction, The presence of missing data or redundant fields in IDS data often affects the training and reliability of the model. The working method of PCA is to map high-dimensional data to low-dimensional space representations through linear projection. To achieve the goal of preserving sample variance while reducing the original data dimension, multiple highly correlated variables are transformed into independent or uncorrelated variables.

Assuming there are m samples in the original dataset, use $x_1, x_2, \dots, x_m$, There are n features in each sample space. If you want to extract the main features or components from the dataset, use n' $(n' < n)$ to represent them. The main calculation process is as follows:

(1) Firstly, we need to standardize the original data.

$$M_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j} \quad , i = 1,2,\dots,m; j = 1,2,\dots,n$$

Wherein $\bar{x}_j = \frac{\sum_{i=1}^{m} x_{ij}}{m}$, $s_j = \sqrt{\frac{\sum_{i=1}^{m}(x_{ij} - \bar{x}_j)^2}{m-1}}$

(2) Calculate the correlation coefficient matrix

$$R = \frac{M^T M}{m-1}$$

(3) Assuming $\lambda_1, \lambda_2, \dots, \lambda_n$ are the correlation coefficients and are the eigenvalues of matrix R. Calculate the eigenvectors of the corresponding units as follows:

$$a_1 = \begin{bmatrix} a_{11} \\ a_{21} \\ \dots \\ a_{n1} \end{bmatrix}, a_2 = \begin{bmatrix} a_{12} \\ a_{22} \\ \dots \\ a_{n2} \end{bmatrix}, \dots, a_n = \begin{bmatrix} a_{1n} \\ a_{2n} \\ \dots \\ a_{nn} \end{bmatrix}$$

(4) Calculate principal components

$$t_i = a_{1i} M_1 + a_{2i} M_2 + \cdots + a_{ni} M_n \, , i = 1,2, \dots, n'$$

Finally, n' principal components are used as new data vectors to replace the original data. PCA algorithm is used for feature dimensionality reduction and effective feature extraction, which can maximize the removal of unimportant features, reduce data redundancy, and ensure the effectiveness of the trained IDS model.

2.3 CNN-LSTM Network Model

Due to the characteristics of IDS system, IDS models require long-term dependencies between extracting spatial features and long-range temporal features[15]. LSTM can effectively solve problems such as vanishing or exploding gradients during long sequence training, and its training effectiveness can be effectively improved as the sequence length increases. The model structure of LSTM includes three gates, namely input gate, forget gate, and output gate[16]. The working principles of these three gates are similar, but their working modes are completely different. The input gate determines which node states to add new information to, the forget gate determines which information node states need to lose, and the output gate determines which information the current state needs to output. In addition, it also includes a memory unit used to maintain long-term connections between features. The following figure shows a part of the network structure of LSTM, where $tanh$ represents the hyperbolic tangent function, The gates in the LSTM structure are composed of dot products and sigmoid functions[17], The output range of $sigmod$ is from 0 to 1,1, indicating the release of all data, 0 indicating no data flow, and the output range corresponds to the proportion of data flow passing through the node.

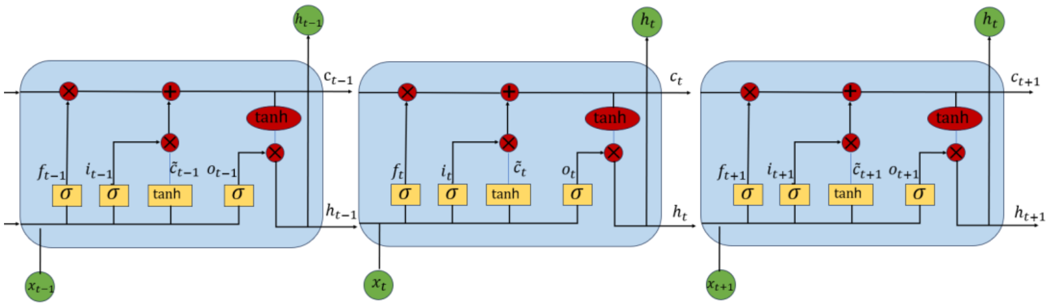


Figure 2 LSTM model framework diagram

Because of its excellent performance in maintaining temporal correlation features, LSTM has been widely applied in the field of deep learning. However, its drawback is that the model suffers from severe spatial data redundancy during the computation process. Because LSTM uses fully connected input to state and state to state transitions without compressing and encoding the spatial data, it extracts too much complex data. To solve this problem, this article innovatively uses the CNN-LSTM network for IDS feature modeling[18]. Taking the input gate as an example, the following are the calculation formulas for LSTM and the innovative CNN-LSTM:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci} \circ C_{t-1} + b_i)$$

$$i_{t(Cov)} = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i)$$

CNN-LSTM incorporates convolution operations in the calculation process of input and output states, to compress redundant space in the data and extract high-quality data[19]. The optimized model combines convolution with LSTM depth, achieving significant results in extracting spatial features of the data.

In addition, because most of the data in IDS is network traffic data, which has special characteristics, this article adjusts the CNN and LSTM structures to adapt to the actual use of IDS, and innovatively proposes a weight adjustment mechanism called CNN-LSTM. The AWD-CNN-LSTM model is a CNN-LSTM neural network that uses weight adjustment techniques to regularize[20]. Its principle is to replace the $W_{hi} * H_{t-1}$ in formula (3) with $(R_t \circ W_{hi}) * H_{t-1}$, introduce dynamic sparsity in the weight matrix W_{hi}, so that the connections between each node are set to zero with a probability of $1 - p$. During the IDS model training phase, the AWD-CNNLSTM network is applied to adjust the weight matrix W between hidden layers to prevent overfitting, which can improve the performance of the model applied to network traffic datasets. The following is the calculation formula for AWD-CNN-LSTM:

$$i_{t(Aw-cov)} = \sigma(W_{xi} * X_t + (R_{it} \circ W_{hi}) * H_{t-1} + W_{ci} \circ C_{t-1} + b_i)$$

$$f_t = \sigma(W_{xf} * X_t + (R_{ft} \circ W_{hf}) * H_{t-1} + W_{cf} \circ C_{t-1} + b_f)$$

$$o_t = \sigma(W_{xo} * X_t + (R_{ft} \circ W_{hi}) * H_{t-1} + W_{co} \circ C_{t-1} + b_o)$$

$$C_t = f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * x_t (R_{ct} \circ W_{hc}) * H_{t-1} + b_c)$$

$$H_t = o_t \circ \tanh(C_t)$$

In the above calculation formula, i_t represents the input gate in the network, f_t represents the forget gate, o_t represents the output gate, C_t represents the memory cell unit, H_t represents hidden state, * represents convolutional operation, and ○ represents product of Ada codes, R is a binary mask matrix.

3. Methodology

The IDS model integrates convolutional neural networks and LSTM, and uses a GP-GAN network to balance abnormal and normal traffic during training. In response to the data redundancy and feature extraction problems in IDS, this paper uses principal component analysis (PCA) algorithm[21]. The following algorithm not only solves the problem of IDS data imbalance, but also avoids data redundancy, extracts effective features from the dataset, and improves the network feature expression ability. The traditional IDS model mainly analyzes and extracts intrusion patterns and attack features, and constructs rule libraries, template libraries, etc., resulting in low detection accuracy and weak generalization ability of existing IDS models. Therefore, it is necessary to combine optimized deep-learning techniques to improve the accuracy of IDS model recognition and classification[22]. The overall framework diagram of IDS in this article is shown in Figure 3. The entire framework mainly includes four parts.

(1) Data preprocessing

Firstly, convert the character features of the original dataset into numerical features, and then normalize all features to balance the weight of each feature in the data, to accelerate the speed of gradient descent to find the optimal solution.

(2) GP-GAN abnormal data oversampling

After data preprocessing in step (1), the training set is divided into rare anomaly type data and other types of data. We use rare anomaly type data as network input and use GP-GAN to generate high-quality data to obtain oversampled data. This process can balance the proportion of abnormal data and normal data.

(3) PCA data dimensionality reduction

Combining GP-GAN processed data with other types of data as a new training set, the test set and the new training set are reduced in data dimension using PCA dimensionality reduction algorithm, and effective features are extracted to reduce redundant data and improve training efficiency.

(4) Train and test the AWD-CNN-LSTM model

The PCA dimensionality reduced data is used as input for the AWD-CNN-LSTM model, and after iterative training and adjusting relevant parameters until the model converges, it can be used for IDS model traffic classification, accurate classification and localization of intrusion data types.

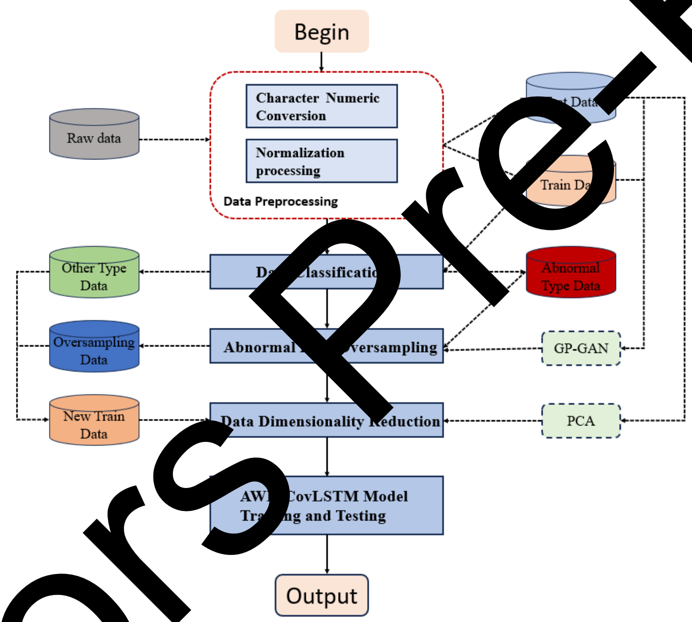


Figure 3 Methodology flowchart

4. Experiment and Analysis.

To evaluate the performance of the improved model proposed in this article, accuracy, precision, recall, and F1 value were selected as the evaluation indicators of the model. Accuracy represents the proportion of correct judgments on the entire sample, Precision represents the proportion of correctly predicted data with true values, Recall refers to predicting the correct proportion among all data with correct true values. The F1 indicator combines the results of Precision and Recall, with a range of 0-1 indicating the best output of the model. Table 1 represents the confusion matrix.

Table 1 Confusion matrix

Predictive Labels	Authentic Labels	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

TP represents the number of correctly classified positive samples, FP represents the number of misclassified positive samples, TN represents the number of correctly classified negative samples, FN represents the number of misclassified negative samples. The definition of confusion matrix can lead to Accuracy The calculation formulas for Precision, Recall, and F1 are:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

The experimental environment was developed using the Win11 operating system environment, and the detailed experimental environment is shown in Table 2.

Table 2 Table of Experimental Environment Configuration

Experimental Environment	Specific Configuration
Operating System	Windows11 64bit
CPU	12th Gen Intel(R) Core(TM) i5-12450H 2.50 GHz
GPU	Intel(R) UHD Graphics
Memory	32GB
Hard Disk	1TB
Development Framework	PyTorch1.8.1
Development Language	Python 3.9.8

This article used the NSL-KDD dataset to validate the effectiveness of the newly proposed IDS model[23], which is widely used and evaluated as the KDD99 dataset in the IDS field, NSL-KDD is an improved version of KDD99, The NSL-KDD dataset has been widely used in the field of IDS for evaluation. This dataset includes two parts: the training set (KDDTrain+) and the testing set (KDDTest+)[24], and the proportions of the two parts are similar. Researchers do not need to verify the partition of the dataset. In addition, the NSL-KDD dataset does not contain redundant records, and each data in the testing set has uniqueness, making the classification and validation results of NSL-KDD convincing. Table 3 shows the distribution of the NSL-KDD dataset[25].

Table 3 Distribution Table of NSL-KDD Dataset

Dataset	Type	Number	Proportion
KDDTrain+	Normal	67343	53%
	Dos	45927	37%

	Probe	11656	9.11%
	U2R	52	0.04%
	R2L	995	0.85%
KDDTest+	Normal	9711	43%
	Dos	7458	33%
	Probe	2421	11%
	U2R	200	0.9%
	R2L	2754	12.1%

Each data in NSL-KDD contains 1 label and 41 features, with a total of 38 numerical features and 3 character features. The types of labels include normal traffic (Normal), denial of service (Dos) attacks, probe attacks, user to root (U2R) attacks, and remote to local attacks (R2L).

4.1 Data Preprocessing

Each data in the NSL-KDD dataset consists of a total of 41 features, of which 3 are character based features and 38 are numerical based features. Before model training, data preprocessing is a crucial step in the experimental process to ensure the accuracy of training and results. The data preprocessing method for this experiment is as follows:

(1) Using One hot encoding to numerically transform character features. The 41 features in the original dataset are mapped to 123 features.

(2) Data normalization, in order to balance the contribution of each feature, this paper uses Min Max normalization to map the features to the range of 0-1, The Min Max calculation formula is as follows:

$$x^* = \frac{x - min}{max - min}$$

Where max is the maximum value of the data, min is the minimum value of the data, x is the input data, x^* represents the normalized data.

(3) Label processing, using One hot encoding to convert 5 types of labels (Normal Maps Dos, Probe, U2R, R2L) to 0-4 to prepare for subsequent calculations.

4.2 PCA Data Dimensionality Reduction

There is significant data redundancy in the IDS features, which not only lowers the process's learning efficiency but also has an impact on training accuracy. Therefore, dimensionality reduction of high-dimensional data is also an important step in the experiment. The experiment uses PCA algorithm to reduce the dimensionality of preprocessed data, and analyzes each principal component after transformation to determine the dimensionality reduction coefficient.

Firstly, let's assume $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ are the eigenvalues of the correlation coefficient matrix R, then the formula for calculating the variance contribution rate of the k-th principal component is:

$$C_i = \frac{\lambda_k}{\sum_{i=1}^{n} \lambda_i} \quad i \in \{1,2,\ldots,n\}$$

The formula for calculating the cumulative variance contribution rate is:

$$Sum_{C_k} = \frac{\sum_{i=1}^{k} \lambda_k}{\sum_{i=1}^{n} \lambda_i} \quad k \in \{1,2,\ldots,n\}$$

As shown in figure 4 below, the Sum_{C_k} curve graph is drawn for the 124 dimensional features of the NSL-KDD training set (KDDTrain+dataset) after data preprocessing. From the graph, it can be seen that the first 80 principal components retain more than 99% of the effective information, so this article sets the dimensionality reduction dimension to 80.

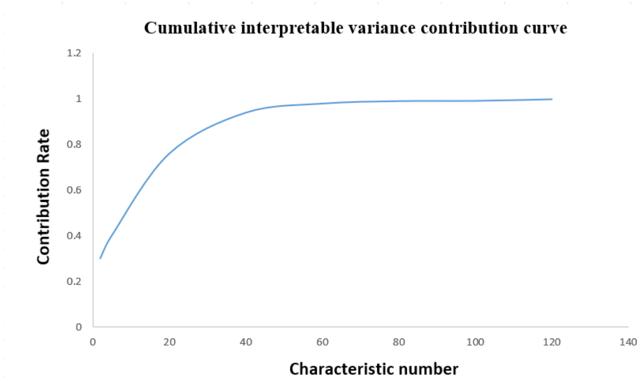


Figure 4 Cumulative interpretable variance contribution curve

The principle of GAN is to generate the optimal solution through the game process between the generator and discriminator. Therefore, the network structure of the generator and discriminator should be balanced. The specific model structure of the GP-GAN implemented in this article is shown in the following figure. The number of input layer nodes for the generator is 80. The number of nodes in the four hidden layers of the generator is 128, 256, 512, and 1024, respectively. The activation function for each hidden layer is ReLU. The number of nodes in the output layer of the generator and the input layer of the discriminator is set to feature number 124 of the NSL-KDD dataset after data preprocessing, and the activation function is Tanh. The discriminator consists of two hidden layers, with a node count of 512 and 256, respectively. Since GAN only needs to determine whether the data is true or false, the output layer of the discriminator only needs one node.

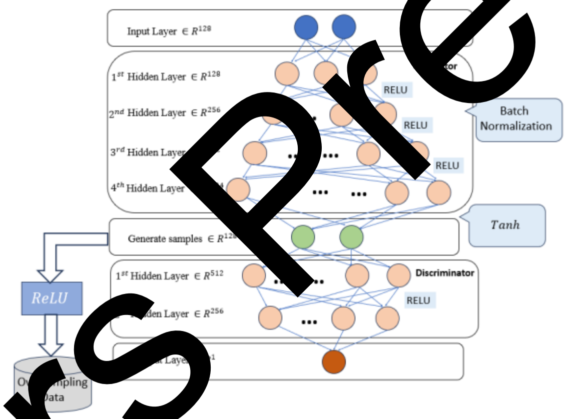


Figure 5 GP-GAN model framework diagram

In the process of neural networks, the low quality of generated samples may be caused by their characteristics, but the training of GANs is often accompanied by special training algorithms. Excellent algorithms can achieve approximate synchronization between the generator and discriminator by controlling the number of alternating training times to avoid problems such as gradient vanishing or pattern collapse. The following are the key training steps for GP-GAN.

$Input: \lambda, \alpha, \beta_1, \beta_2, n$

$Output: \omega, \delta$

1. While δ has not converged do;
2. For t = 1,2, … , n do
3. For i = 1, … , m do
4. $Sample\ x\sim P_r\ , z\sim P_z\ , \varepsilon\sim(0,1)$
5. $\tilde{x}\ \leftarrow\ G_\theta(z)$
6. $\hat{x} \leftarrow (1-\varepsilon)\tilde{x}$

7. $L_D^i \leftarrow D_w(\tilde{x}) - D_w(x) + \lambda(\|\nabla_{\hat{x}} D_w(\hat{x})\|_2 - 1)^2$

8. $End\ for;$

9. $w \leftarrow Adam(\nabla_\omega \frac{1}{m}\sum_{i=1}^{i=m} L_D^i, \alpha, \beta_1, \beta_2)$

10. $End\ for;$

11. $Sample\ \{z^{(i)}\}_{i=1}^{i=m} \sim P_z$

12. $\delta \leftarrow Adam(\nabla_\delta \frac{1}{m}\sum_{i=1}^{i=m} -D_\omega(G_\delta(z), \delta, \alpha, \beta_1, \beta_2))$

13. $End\ while;$

The following is a description of the parameters used in the above training process, as well as the parameter settings. Please refer to Table 4 for details.

Table 4 GP-GAN training model parameter configuration table

Parameter	Description	Settings
ω	Discriminator parameters	Output Value
δ	Generator parameters	Output Value
α	Learning rate	0.001
β_1, β_2	Optimizer parameters	0.5/0.9
λ	Gradient penalty coefficient	10

Train GP-GAN using the above algorithm until the model converges. In response to the issue of imbalanced NSL-KDD dataset, this article focuses on the two attack types (R2L) with the lowest proportion in the KDDTrain+training set, U2R generates high-quality samples, and the data distribution before and after oversampling is shown in Table 5.

Table 5 KDDTrain+data distribution table before and after oversampling using GP-GAN

Type	Number before oversampling	Number after oversampling
Normal	67343	67343
Dos	45927	45927
Probe	11656	11656
U2R	52	10052
R2L	995	10995

In the previous section, the data preprocessing module has been discussed, which converts the character-based features of the original dataset into numerical features and uses the GP-GAN network to balance the proportion of abnormal data and normal data. In the following section, the AWD-CNN-LSTM model will be introduced in detail. The following diagram shows the structure of the AWD-CNN-LSTM model.

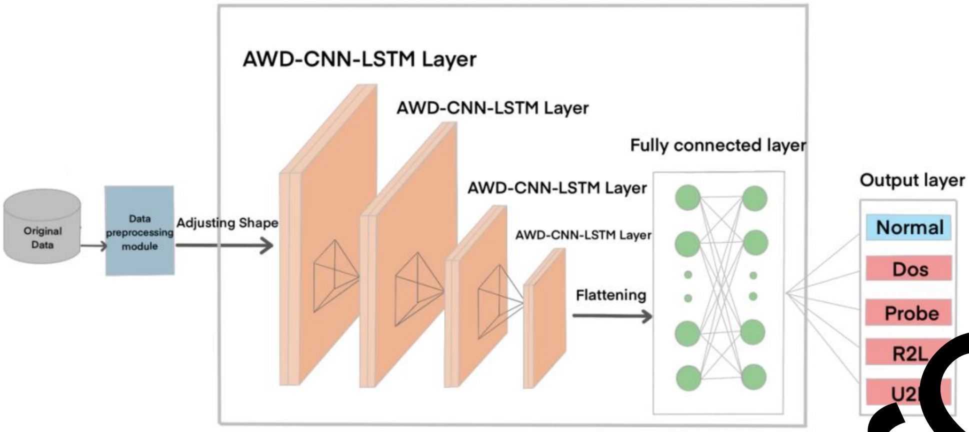


Figure 6 AWD-CNN-LSTM model framework diagram

After data preprocessing, the original data is converted from character type to numerical type, and the spatial representation of each data is 1 * 122. After dimensionality reduction using PCA algorithm, the spatial representation of each data is 1 * 100. Next, according to the network structure of AWD-CNN-LSTM, the 1 * 100 dimension must be mapped to 2 * 2 * 5 * 5 data dimensions to extract features from it.

As shown in Figure 6, in the neural network layer section, it is necessary to reduce data redundancy and effectively extract high-quality spatial features. Therefore, this paper sets up four layers of AWD-CNN-LSTM to extract spatial features from high-dimensional data. By inputting the state of LSTM nodes and adding convolutional computation to the full connections between each node state, high-quality data features are extracted.

After the above processing, high-quality data space features have been extracted, and the next step is to classify the traffic data. Flatten the extracted spatial features and input them into three fully connected layers for traffic classification. In order to improve training efficiency and prevent overfitting, the fully connected layer activation function is set to BN and RELU. Since the NSL-KDD dataset has 5 types of labels, the number of output nodes in the final fully connected layer is set to 5.

4.3 Experiment Results

Train WGAN-GP and GAN separately and generate data. Among them, The parameters used by WGAN-GP have been given in the previous text, The dimension and learning rate of the random noise vector used by GAN are consistent with those of GP-GAN, with values of 100 and 0.001, respectively. This article uses root mean square error (RMSE) to measure the degree of fit between generated samples and real samples. The convergence curves of GP-GAN and GAN training are shown in Figure 7.

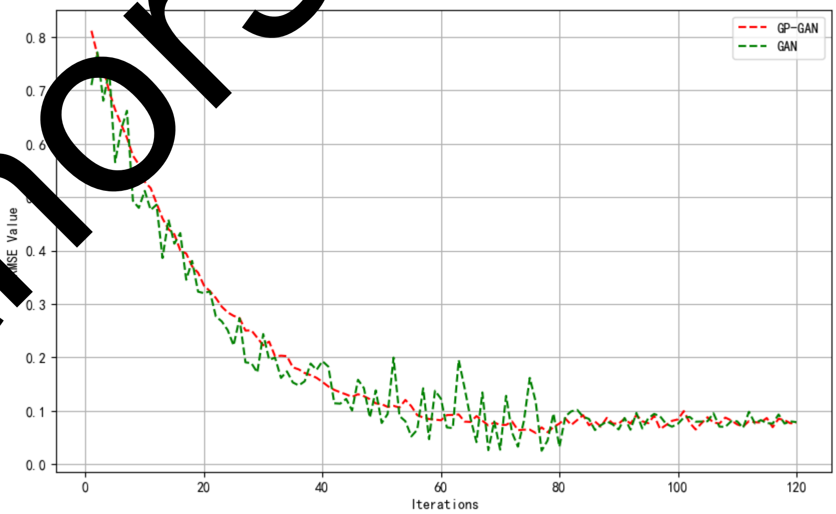


Figure 7 Comparison chart of convergence curves between GAN and GP-GAN

It can be seen from the graph that firstly, the convergence curve of GP-GAN is smoother compared to that of GAN, without significant oscillations. In addition, the convergence curve trends of the two models are roughly the same, reaching a convergence state in about the 90th iteration. After convergence, the root mean square difference of GP-GAN is significantly smaller than that of GAN, indicating that the fitting degree of the generated data of GP-GAN is higher than that of GAN. The reason for the above experimental results is that the GP-GAN network has added a gradient penalty term to avoid gradient disappearance. This condition can effectively limit the growth rate of the function, make the training of the model more stable, and improve the quality of generated data.

Due to the severe imbalance of IDS data, it can lead to model bias and ultimately result in suboptimal training performance of the IDS model. This article adds a weight adjustment mechanism to the CNN-LSTM model to make it more suitable for recognizing IDS data. The AWD-CNN-LSTM models used are listed in Table 6. Now, the same optimizer and learning rate are used for CNN-LSTM, and the two models are applied to the NSL-KDD dataset. The experimental results are shown in Table 6. It can be concluded that AWD-CNN-LSTM has advantages in accuracy, and overall The F1 score outperforms CNN-LSTM in all three important indicators, indicating that the improved AWD-CNN-LSTM model is effective.

Table 6 Comparison table of experimental results before and after CNN-LSTM improvement

Model	Accuracy	Precision	F1-Score
CNN-LSTM	0.687	0.709	0.7[illegible]
AWD-CNN-LSTM	0.712	0.863	0.780

To verify the performance of the AWD-CNN-LSTM model, this paper compares deep learning and some machine learning algorithms. Reference [] applied traditional machine learning algorithms to IDS and conducted simulation tests on the NSL-KDD dataset. The RF The experimental results of SVM and KNN algorithms are part of the comparative experiment in this article. Reference [] preprocessed the dataset and inputted it into a hybrid model of MSCNN and LSTM, and achieved excellent results in the experiment. Reference [] uses stacked denoising autoencoder (SDAE) to learn the features of the dataset and input them into ELM to obtain classification results. Reference [] applies CNN to IDS and achieves the goal of improving detection accuracy through cross-entropy loss function. Reference [] proposes a network intrusion detection model that integrates bidirectional gated recurrent unit (CNN-BIGRU) and attention mechanism to improve the feature extraction ability and classification accuracy of network IDS. The performance comparison of machine learning, deep learning algorithms, and the newly proposed AWD-CNN-LSTM model before oversampling is shown in Figure 8 and 9.

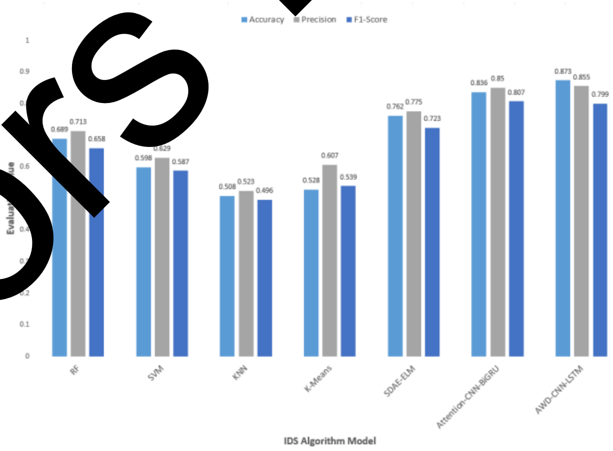


Figure 8 Comparison chart of experimental results of various algorithms

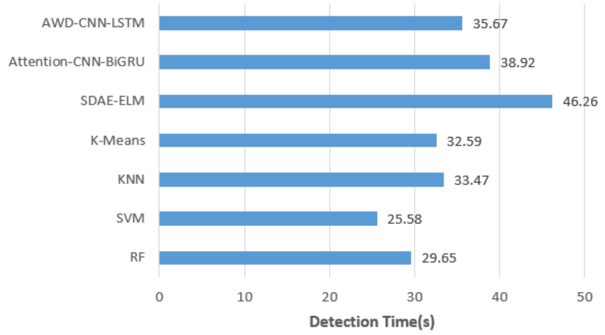


Figure 9 Comparison chart of detection efficiency of various algorithms

It can be seen from Figure 8 that from the accuracy In terms of F1 evaluation value and other aspects, SVM algorithm has the best performance among traditional machine learning algorithms, while SDAE-ELM. The performance of deep learning models such as Attention CNN BiGRU is superior to machine learning algorithms, demonstrating the superiority of deep learning models in the field of intrusion detection. In this paper, the newly proposed model in Accuracy, The F1 Score and Precision evaluation values are superior to other deep learning models, reflecting the powerful performance of AWD-CNN-LSTM in extracting spatial features from data. From the performance analysis of detection efficiency, the newly proposed model (AWD-CNN-LSTM) in this article is relatively close to other machine learning algorithms in detection efficiency and has the highest efficiency in detection time compared to other deep learning models. Based on the above analysis, the performance of the model in this article is the best.

In addition, to verify the effectiveness of the oversampling (GP-GAN) method, this paper uses the AWD-CNN-LSTM model before and after oversampling and the NSL-KDD dataset as a comparative experiment to demonstrate the improvement of oversampling operation on the performance of the IDS model. As mentioned earlier, U2R and R2L are the two least abnormal data types, so the oversampled traffic data types in this article are U2R and R2L. To demonstrate the effectiveness of the oversampling method provided in this article, Table 7 shows U2R The growth of various evaluation indicators of R2L and population samples before and after oversampling.

Table 7 Comparison table of experimental results before and after oversampling

Type		Evaluating Indicator		
		Precision	Recall	F1-Score
R2L	Before Oversampling	0.023	0.015	0.027
	After Oversampling	0.168	0.165	0.164
	Growth Rate	0.145	0.150	0.137
U2R	Before Oversampling	0.668	0.153	0.234
	After Oversampling	0.806	0.232	0.358
	Growth Rate	0.148	0.079	0.124
Overall	Before Oversampling	0.789	0.781	0.790
	After Oversampling	0.812	0.795	0.850
	Growth Rate	0.023	0.014	0.09

From Table 7, it is evident that the extremely imbalanced distribution of the NSL-KDD dataset results in the model being insensitive to detecting rare attack types. This article uses GP-GAN to generate rare attack type data, balancing the distribution of different types of data and effectively reducing the bias problem of the model. For U2R type detection data, its precision, recall The values of F1 Score have increased by 14.8%, 7.9%, and 12.4%, respectively. For the R2L attack type, these three indicators have increased by 14.5%, 15%, and 13.7%, respectively. For the overall sample, oversampling increased the precision by 2.3% and the recall by 1.4%, F1 score increased by 6%.

5. Conclusion

The CNN-LSTM model was utilized in the field of intrusion detection in this article, and an innovative AWD-CNN-LSTM was suggested based on the IDS model's properties. Due to the extremely imbalanced distribution of the NSL-KDD dataset, the model is not sensitive to detecting rare attack types. In this article, GP-GAN is used to generate rare attack type data, balancing the distribution of different types of data, and effectively alleviating the detection problem caused by imbalanced intrusion detection data. The optimized IDS model presented in this article performs exceptionally well and is capable of effectively identifying different types of attacks, as confirmed by experimental verification. The following stage involves applying the GP-GAN model to concentrate on producing a specific percentage of rare type data to maximize model performance.

Compliance with Ethical Standards

Competing Interests

Baoguo Liu declares that he has no conflict of interest. Eric B. Blancaflor declares that he has no conflict of interest. Mideth Abisado declares that he has no conflict of interest.

Funding: This study has no funding.

References

[1]Denning D.E. An Intrusion-Detection Model[J]. Institute of Electrical and Electronics Engineers (IEEE), (2): 222-232.
[2]Arwa Aldweesh,Abdelouahid Derhab,Ahmed Z. Emam. Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues[J]. Elsevier Bv, [illegible]020, 189: 105124.
[3]Reda M. Elbasiony,Elsayed A. Sallam,Tarek E. El[illegible] et al. A hybrid network intrusion detection framework based on random forests and weighted k-means[J]. Elsevier Bv, 201[illegible], [illegible]4(4): 753-762.
[4]Shubair A.,Ramadass Sureswaran,Altyeb Alty[illegible]-Altah[illegible]. kENFIS: kNN-based evolving neuro-fuzzy inference system for computer worms detection[J]. IOS Press, [illegible], 26(4): 1893-1908.
[5] Zhang Anlin, Zhang Qikun, Huang Daoying, etc Intrusion detection model based on CNN and BiGRU fusion neural network [J] Journal of Zhengzhou University (Engineering Edition), 2022, 43(03): 37-43.
[6]Acharya Toya,Annamalai Annamalai,C[illegible]ikha Mohamed-F. Efficacy of Bidirectional LSTM Model for Network-Based Anomaly Detection[C]//202[illegible] IEEE [illegible]th Symposium on Computer Applications & Industrial Electronics (ISCAIE): IEEE, 2023: 336-341.
[7] Tao Zhiyong, Yan Minghao, L[illegible] Ying Channel coding closed set recognition based on temporal convolutional networks [J] Journal of Huazhong University of Science and Technology (Natural Science Edition), 2022, 50(03): 12-17.
[8]Ammar Aldallal. Toward Efficient Intrusion Detection System Using Hybrid Deep Learning Approach[J]. Mdpi Ag, 2022, 14(9): 1916.
[9]MHMOOD HA[illegible]. A Novel Approach to Network Intrusion Detection System Using Deep Learning for Sdn: Futuristic Approach[J]. Elsevier Bv, 2022.
[10]Zheng [illegible]. Network Intrusion Detection Model Based on Convolutional Neural Network[C]//IEEE: IEEE, 2021.
[11]Yam[illegible]thi N[illegible]asimha Rao,Kunda Suresh Babu. An Imbalanced Generative Adversarial Network-Based Approach for Network Intrusion Detection in an Imbalanced Dataset[J]. Mdpi Ag, 2023, 23(1): 550.
[12][illegible]moos M[illegible]ammad,García Antonio,Alkhanafseh Mohammad,等. A New Data Balancing Approach based Generative Adversarial Network for Network Intrusion Detection System[Z]: MDPI AG, 2023.
[13]Zhang Junjie,Zhao Ying. Research on Intrusion Detection Method Based on Generative Adversarial Network[C]//IEEE: IEEE, 2021.
[14]Hemavathi K.,Latha R.. Conditional Generative Adversarial Network with Optimal Machine Learning Based Intrusion Detection System[C]//IEEE: IEEE, 2023.

[15]Farkhady Roya-Zareh,Majidzadeh Kambiz,Masdari Mohammad,等. A novel feature selection algorithm for IoT networks intrusion detection system based on parallel CNN-LSTM model[Z]: Research Square Platform LLC, 2023.
[16]Ran Ziyong,Zheng Desheng,Lai Yanling,等. Applying Stack Bidirectional LSTM Model to Intrusion Detection[J]. Computers, Materials and Continua (Tech Science Press), (1): 309-320.
[17]Saurabh Kumar,Sood Saksham,Kumar P.-Aditya,等. LBDMIDS: LSTM Based Deep Learning Model for Intrusion Detection Systems for IoT Networks[C]//IEEE: IEEE.
[18]Jiawei Du,Kai Yang,Zhentao Huang,等. Research on Intrusion Detection Algorithm Based on Optimized CNN-LSTM[C]//IEEE: IEEE.
[19]Rudra Bhawna,Agrawal Vivek-Kumar. Deep learning for network security: a novel GNN-LSTM-based intrusion detection model[J]. Inderscience Publishers, 2023, (1).
[20]Harshitha T-Sai. Intrusion Detection and Prevention Using CNN-LSTM[J]. Jaipur Innovative Research Publication Center, (2): 1-6.
[21]Karamollaoğlu Hamdullah,Yücedağ İbrahim,Doğru İbrahim-Alper. A Hybrid PCA-MAO Based LSTM Model for Intrusion Detection in IoT Environments[Z]: Research Square Platform LLC.
[22]Altunay Hakan-Can,Albayrak Zafer. A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks[J]. Elsevier BV: 101322.
[23]Solanki Surbhi,Gupta Chetan,Rai Kalpana. A Survey on Machine Learning based Intrusion Detection System on NSL-KDD Dataset[J]. Foundation of Computer Science, (30): 36-39.
[24]Shehadeh Aseel,Altaweel Hanan,Qusef Abdallah. Analysis of Data Mining Techniques on KDD-Cup'99, NSL-KDD and UNSW-NB15 Datasets for Intrusion Detection[C]//IEEE: IEEE.
[25]Walling Supongmen,Lodh Sibesh. Performance Evaluation of Supervised Machine Learning Based Intrusion Detection with Univariate Feature Selection on NSL KDD Dataset[Z]: Research Square Platform LLC.