Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.061206

ARTICLE

SA-ResNet: An Intrusion Detection Method Based on Spatial Attention Mechanism and Residual Neural Network Fusion

Zengyu Cai[1,*], Yuming Dai[1], Jianwei Zhang[2,3,*] and Yuan Feng[4]

[1]School of Computer Science and Technology, Zhengzhou University of Light Industry, Zhengzhou, 450066, China
[2]College of Software Engineering, Zhengzhou University of Light Industry, Zhengzhou, 450066, China
[3]Faculty of Information Engineering, Xuchang Vocational Technical College, Xuchang, 461000, China
[4]School of Electronic Information, Zhengzhou University of Light Industry, Zhengzhou, 450066, China

*Corresponding Authors: Zengyu Cai. Email: mailczy@163.com; Jianwei Zhang. Email: mailzjw@163.com

Received: 19 November 2024; Accepted: 24 February 2025; Published: 16 April 2025

ABSTRACT: The rapid development and widespread adoption of Internet technology have significantly increased Internet traffic, highlighting the growing importance of network security. Intrusion Detection Systems (IDS) are essential for safeguarding network integrity. To address the low accuracy of existing intrusion detection models in identifying network attacks, this paper proposes an intrusion detection method based on the fusion of Spatial Attention mechanism and Residual Neural Network (SA-ResNet). Utilizing residual connections can effectively capture local features in the data; by introducing a spatial attention mechanism, the global dependency relationships of intrusion features can be extracted, enhancing the intrusion recognition model's focus on the global features of intrusions, and effectively improving the accuracy of intrusion recognition. The proposed model in this paper was experimentally verified on the NSL-KDD dataset. The experimental results show that the intrusion recognition accuracy of the intrusion detection method based on SA-ResNet has reached 99.86%, and its overall accuracy is 0.41% higher than that of traditional Convolutional Neural Network (CNN) models.

KEYWORDS: Intrusion detection; deep learning; residual neural network; spatial attention mechanism

1 Introduction

Internet continues to develop and become more widespread, network security has become increasingly important in modern society. As technology advances, various intrusion techniques are continuously updated. Traditional network security technologies struggle to counter the malicious intrusions posed by new network viruses and fail to meet the demands of contemporary societal development. Intrusion detection technology is a crucial safeguard for network security, necessitating the development of new methods to enhance detection effectiveness. With the recent advancements and maturation of deep learning technologies, they have provided fresh perspectives for the research and development of intrusion detection systems. Deep learning-based intrusion detection has emerged as a prominent topic and trend in research.

Intrusion detection systems (IDS) can be categorized into two main methods: anomaly detection and misuse detection technologies [1]. In 1980, Anderson et al. [2] first introduced the concept of computer security threat monitoring and surveillance. Traditional machine learning techniques have been utilized in intrusion detection for several decades. In 1986, Dorothy Denning from Georgetown University and Peter Neumann from SRI developed the first real-time intrusion detection expert system which was a rule-based

and statistical intrusion detection expert system [3], capable of auditing and analyzing log files and host access information using naive Bayes and decision tree methods to detect abnormal behavior and prevent human intrusions. Wang et al. [4] proposed an intrusion detection algorithm utilizing a single-class Support Vector Machine (SVM) combined with a Gaussian mixture model, which enhanced the detection rate. Ali et al. [5] introduced a bagging algorithm that integrates data analysis techniques with four robust machine learning ensemble methods, effectively reducing Mean Square Error (MSE) and Mean Absolute Error (MAE).

In the rapidly evolving era of big data, new forms of attacks emerge daily. Traditional machine learning methods are shallow learning models that lack the ability to autonomously learn features and do not adequately meet current requirements. These methods continue to face challenges, including unsatisfactory feature extraction results and low detection accuracy. The advancement of deep learning offers significant potential for the field of intrusion detection. References [6–10] indicate that deep learning algorithms, including Auto Encoders (AE), CNN, Recurrent Neural Networks (RNN), Long Short-Term Memory networks (LSTM), and Generative Adversarial Networks (GAN), have been extensively applied in intrusion detection. Lee et al. [11] proposed a network anomaly detection model utilizing a stacked sparse AE and CNN, trained in a semi-supervised manner. This model extracts new feature vectors using SSAE and detects network anomalies with DeepCNN. Zhao et al. [12] proposed an intrusion detection method that employs semi-supervised federated learning through knowledge distillation, combining convolutional neural networks to extract deep-level features. This approach achieves improved detection performance and reduced communication overhead. Deore et al. [9] proposed an intrusion detection model based on a RNN classifier for feature reduction, integrating correlation and information gain via the RNN algorithm. Yang Xiaowen et al. [13] proposed a network intrusion detection model that integrates CNN-BIGRU (Convolutional Neural Network-Bi-directional Gated Recurrent Unit) with an attention mechanism, outperforming ensemble models in network traffic feature extraction. Nonetheless, challenges remain in balancing datasets and detection rates of the models. Wang et al. [14] proposed a spiral convolution LSTM fusion model for network intrusion detection, employing a method that incorporates spiral convolution, a double-layer LSTM network, and a classifier through one-hot encoding and normalized preprocessing of the dataset. Alrayes et al. [15] introduced an intrusion detection system based on CNN and a channel attention mechanism, achieving an accuracy of 99.728% on the NSL-KDD dataset and significantly enhancing the performance of intrusion detection. To address the issue of low accuracy in multi-class classification within intrusion detection, Sun Hongzhe et al. [16] proposed proposed the Attention-BiTCN (BiDirectional Temporal Convolutional Network) model, which effectively extracted the deep and global temporal characteristics of network traffic data through the two-way sliding window method and attention mechanism, and significantly improved the multi classification performance.

As deep learning technologies continue to advance rapidly, a growing number of sophisticated methods are being applied to intrusion detection. In particular, Transformer models, Graph Neural Networks (GNNs), and Generative Adversarial Networks (GANs) have shown remarkable performance in a variety of tasks, presenting novel strategies for enhancing intrusion detection systems. Initially developed for natural language processing tasks, the Transformer model has achieved widespread Notice as a result of its powerful Aptitude for model long-range dependencies. In recent years, researchers have started exploring the application of Transformers to intrusion detection tasks, aiming to fully leverage critical patterns in network traffic and thereby improve detection accuracy and robustness. Wu et al. [17] proposed a Robust Transformer-based Intrusion Detection System (RIDS) that reconstructs feature representations to balance dimensionality reduction with feature preservation in imbalanced datasets. It uses positional embedding techniques to link sequence information across features and leverages the Transformer model to derive low-dimensional feature

representations from high-dimensional raw data. The approach was tested on the CICIDS2017 and CIC-DDoS2019 datasets, with the results highlighting notable improvements in both accuracy and efficiency for intrusion detection. Safi et al. [18] leveraged the parallel processing capabilities of Transformer neural networks to accelerate the learning process, thus improving the detection rate of malicious attacks. Graph neural networks excel at processing graph-structured data. In intrusion detection tasks, network traffic data exhibits graph-structured features, with each packet represented as a node and the links between nodes representing their relationships. Graph neural networks capture the relationships between nodes, enhancing the recognition ability of intrusion detection systems for complex patterns. Friji et al. [19] proposed a framework based on GNNs that uses the graph structure to classify communication flows by assigning malicious scores, offering a new direction for research in intrusion detection systems. GANs offer a novel solution for enhancing data diversity and generating challenging intrusion samples. By training a generator and discriminator, GANs can generate samples resembling real attack behaviors, effectively enhancing the training data of intrusion detection models and improving their generalization and resilience to interference. Seo et al. [20] proposed a GAN-based Intrusion Detection System (GIDS) that accurately detects attack types using known training data and utilizes randomly generated fake data to identify unknown attacks, achieving an average accuracy of 98%. This study shows that GANs can enhance the adaptability of detection systems when confronting unknown attack types.

In summary, to tackle the current network security challenges in the network environment, along with the low accuracy and high false alarm rates associated with traditional networks, Building on existing research, this article proposes an intrusion detection method that combines a spatial attention mechanism with residual neural network fusion. By incorporating a spatial attention mechanism, the model dynamically focuses on regions of the input data that are most relevant to the task. By assigning varying weight values to different spatial positions, the model enhances its ability to capture key information while suppressing the influence of irrelevant or redundant data, thereby effectively improving the accuracy of the network intrusion detection system.

2 Related Work

Currently, deep learning has increasingly been utilized in the field of intrusion detection, becoming a mainstream approach that significantly enhances the protection capabilities of intrusion detection systems. The method proposed in this article is based on the fusion of a spatial attention mechanism and a residual neural network. The spatial attention mechanism is integrated into the deep residual network to learn the temporal correlations of network data and improving its focus on intrusion traffic features.

2.1 Convolutional Neural Network

CNN is a type of neural network inspired by biological research, characterized as a deep feedforward network with a convolutional structure. CNNs mimic the biological visual system by inputting images into the network and extracting their features. By employing local connections and weight sharing, the network becomes easier to optimize and reduces the risk of overfitting.

Convolutional neural networks are widely utilized in image and text processing, effectively extracting features from images. CNN primarily extracts features from input data, utilizing them for classification, detection, and recognition tasks. In the realm of intrusion detection, the challenge is fundamentally a classification problem, where multi-classification models trained via supervised learning are applied to network traffic data for prediction.

2.2 Deep Residual Neural Network

Similar to RNNs, CNNs also experience issues such as the vanishing gradient problem and network degradation. This limitation prevents the model from being sufficiently deep and results in a small number of training parameters, which in turn hinders the achievement of higher accuracy. In 2016, He et al. [21] proposed residual neural networks to tackle the "network degradation phenomenon" that occurs during training. By introducing a deep residual learning framework, the issues of vanishing and exploding gradients associated with increased network depth have been significantly mitigated.

The residual module allows the network to maintain performance as depth increases by stacking layers (identity mappings) on top of a shallow network. Residual neural networks primarily consist of basic residual blocks that utilize "shortcut connections" to compute weights and threshold partial derivatives during model backpropagation. This connection method reduces the complexity of model training, and its structure is illustrated in Fig. 1. ResNet learns residual functions, which at that time acted as identity mappings without introducing additional parameters or computational complexity when residual functions were ineffective.

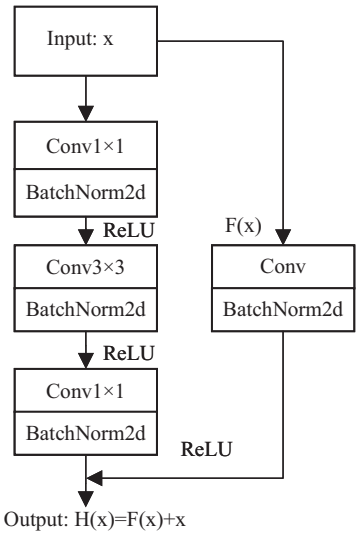

Figure 1: Residual module structure

2.3 Spatial Attention Mechanism

In recent years, spatial attention mechanisms have made significant advancements in computer vision, particularly excelling in image recognition and object detection tasks. The spatial attention mechanism creates attention maps by analyzing the spatial positioning of features, applying convolution operations to emphasize important spatial information, and assigning appropriate weights. The channel attention mechanism differs from the spatial attention mechanism in that the latter prioritizes spatial location over feature data. The spatial attention mechanism dynamically focuses on regions within the input data most relevant to the task by assigning varying weights to different spatial positions. This approach sharpens the model's ability to discern key details while minimizing the prominence of features that are irrelevant or pleonastic.

A key challenge in intrusion detection tasks is the prevalence of redundant information and noise in network traffic. Spatial attention mechanisms can dynamically adjust feature map weights, automatically focusing on critical features associated with intrusion behavior, thereby enhancing detection accuracy. While the application of spatial attention mechanisms in computer vision has made notable progress, as

demonstrated in reference [22], where spatial attention mechanisms are used to fuse multi-scale features and capture information across different levels, thereby fully integrating low and high-level features to enhance segmentation performance. Reference [23] employs a spatial attention mechanism to focus on key regions, enhancing anomaly detection tasks, and combines LSTM to develop a network intrusion detection method that significantly improves system performance in complex attack scenarios. Xu et al. [24] proposed the unsupervised model of tssan, which effectively extracts temporal and spatial features by combining the temporal spatial attention mechanism, and significantly improves the detection performance of rare attack types by extracting common features from unlabeled data.

In this study, it incorporated the spatial attention mechanism into intrusion detection tasks, combining it with a residual neural network to boost the model's focus on key features in network traffic. The advantages of this method in improving detection accuracy and reducing false alarm rates were validated through experiments. The thought process of spatial attention mechanism is as follows:

Firstly, the input feature map F with a size of H × W × C is subjected to channel dimension global max pooling and global average pooling, resulting in two H × W × 1 feature maps;

Secondly, The results obtained through global max pooling and global average pooling are concatenated by channels to obtain feature sizes of H × W × 2;

Finally, perform a 1 × 1 convolution operation on the concatenated result to obtain the spatial attention weight matrix Ms through the activation function.

The spatial attention weight matrix Ms can be expressed as:

$$M_s(F) \in R^{H,W} \tag{1}$$

Generate feature maps using pooling methods in the channel dimension:

$$F^s_{\text{avg}} \in R^{1\times H\times W} \tag{2}$$

$$F^s_{\max} \in R^{1\times H\times W} \tag{3}$$

In summary, the calculation formula for spatial attention is as follows:

$$M_s(F) = \sigma\left(f^{1\times 1}\left([\text{AvgPool}(F);\text{MaxPool}(F)]\right)\right) = \sigma\left(f^{1\times 1}\left[F^2_{\text{avg}};F^2_{\max}\right]\right) \tag{4}$$

3 Proposed Model

Traditional convolutional neural network algorithms reduce model complexity by utilizing hierarchical feature sharing to extract traffic characteristics from the network. As the depth of the network increases, the model may overfit, resulting in a drop in performance. This article presents an intrusion detection system based on CNNs, integrating deep residual networks and spatial attention mechanisms to construct the SA-ResNet model. This model not only prevents degradation and loss of accuracy but also enhances adaptability by processing global dependencies in sequential data through the spatial attention mechanism, resulting in improved training accuracy.

The intrusion detection architecture based on SA-ResNet constructed in this article mainly consists of four parts: data preprocessing, Res-CNN feature extraction, spatial attention mechanism, and output module, as shown in Fig. 2.

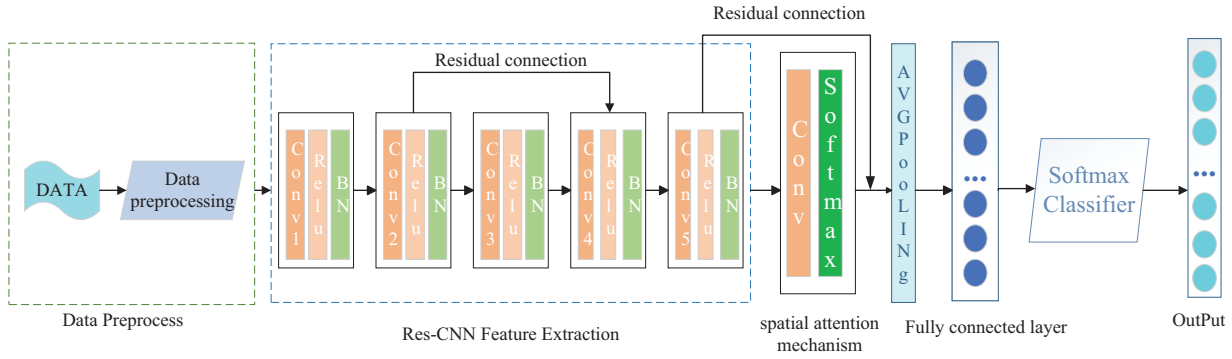

Figure 2: Based on the SA-ResNet intrusion detection model structure

(1) **Data Preprocessing:** Employ one-hot encoding to preprocess the data, ensuring it meets the input requirements of the neural network.

(2) **Res-CNN Feature Extraction:** Extract features from the input data by constructing a residual convolutional neural network.

(3) **Spatial Attention Mechanism:** After the residual neural network calculates the weights, the data is processed in the spatial attention mechanism module for further calculation and optimization.

(4) **Output:** Following the operations of the convolutional residual and attention modules, the outputs from the residual convolution and self-attention modules are combined and processed, followed by adaptive average pooling. Finally, the integrated data is mapped from the previous feature space to the sample label space for classification by the classifier, outputting the data as a five-class classification task through an activation function.

3.1 Data Preprocessing

Data preprocessing is a critical step in developing an intrusion detection system, primarily aimed at preparing data for testing and training. However, the intrusion detection dataset cannot be utilized immediately after downloading; preprocessing is necessary to address quality issues within the dataset. The format of the processed dataset facilitates effective transmission of data to the neural network.

Network intrusion detection involves classifying traffic within a network, as computers can only process numerical data. The original data traffic must be preprocessed to transform it into a format suitable for input into a neural network, based on its characteristics. Subsequently, deep learning methods are employed to extract features for training, culminating in the classification of various types of abnormal traffic.

This article employs one-hot encoding to transform all discrete features into numerical representations. For instance, class labels and network types are digitized, as illustrated in Table 1.

The varying ranges of values for different features can impact the deep learning process, leading to uneven contributions from certain features to the results. Consequently, a standard normalization technique is utilized to preprocess the dataset, guaranteeing that the values of each feature are confined within the [0, 1] range. This normalization aligns all data to a uniform order of magnitude, thereby minimizing the impact on model training results. The normalization formula is expressed as follows:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{5}$$

Table 1: Numerical processing

Class label	Code
Normal	[1,0,0,0,0]
Probe	[0,1,0,0,0]
Dos	[0,0,1,0,0]
U2R	[0,0,0,1,0]
R2L	[0,0,0,0,1]

3.2 Feature Extraction Based on ResNet-CNN

This article develops an intrusion detection model based on SA-ResNet, comprising five convolutional layers and two residual connections. The convolution kernel size for each layer is set to three, with both the stride and padding set to one. Considering the size characteristics and output requirements of the preprocessed data, the network data undergoes activation and regularization through the activation function for each convolutional layer. In addition to basic convolution operations, the extraction of features also incorporates the application of activation functions. Introducing activation functions enhances the network's ability to express non-linear relationships, enabling it to learn more complex functional dependencies. Through non-linear transformations, the network can capture and represent additional dimensions and features of the input data, thereby improving its feature representation capabilities. Common activation functions include the ReLU, Sigmoid, and Tanh functions. This article selects the ReLU activation function for processing the neural networks.

This model incorporates two residual connections. The first residual connection links the outputs of the second and fourth layers, while the second connects the outputs of the fifth layer with the spatial attention mechanism module. The introduction of residual connections allows the network to connect across layers, facilitating gradient propagation to shallower layers. This enhances feature extraction, accelerates network training, and improves model accuracy.

The following provides a brief overview of the calculation steps involved in convolution during the feature extraction process:

(1) Assuming the input of the convolutional layer is, where is the feature quantity at time j. If the output of the i-th convolutional layer is, then the corresponding output of the j-th convolutional kernel is. The output of the input processed by the i-th convolution kernel of the convolutional layer is:

$$x_j^l = s\left(\sum_i x_i^{n-1} \in M_j^{X_i^{l-1}} * W_{ij}^l + b_j^l\right) \tag{6}$$

(2) The pooling layer performs statistical calculations on the feature maps generated by the convolutional layer to preserve the most effective information in the network traffic data. The general form of sampling is:

$$x_j^{i'} = s\left(\beta_j^{i-1} D\left(x_j^{i-1} + b_j^i\right)\right) \tag{7}$$

Among them, D is the sampling function and the β_j^{i-1} is the weight.

3.3 Spatial Attention Mechanism Module

Our proposed spatial attention mechanism is not a mere replication of existing methods but an adaptation that analyzes the specific spatial position information of network traffic features. By conducting a thorough analysis of network traffic data, we have identified key spatial dependencies that traditional attention mechanisms may overlook. Our mechanism assigns varying weight values to different spatial positions based on a novel learning algorithm that identifies and emphasizes the most critical areas for intrusion detection. Assigning different weight values to various spatial positions boosts the model's capacity to grasp essential details while diminishing the impact of unrelated or repetitive information, thus enhancing the model's predictive efficiency and precision.

To capture the spatial relationships between features, a convolution operation is first performed on the input feature map. Next, the softmax layer normalizes the local features extracted from the convolutional layer, associating each feature with a weight and converting each element's value into a probability between 0 and 1. Finally, the output of the softmax layer is multiplied by the input feature values to obtain a weighted feature map. The structure of the spatial attention mechanism is illustrated in Fig. 3.

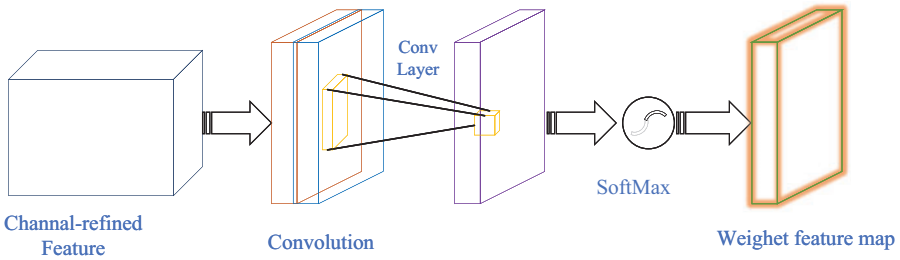

Figure 3: Spatial attention mechanism

3.4 Output Module

(1) Adaptive average pooling

The input data for the fully connected layer is a one-dimensional array of scalar feature values, where each element represents an input feature corresponding to a neuron. It functions as a classifier within the neural network, mapping the learned distributed feature representations to the sample label space for classifying network attacks.

(2) Fully connected layer

After extracting data features, the spatial dimensions of the features are reduced. The connections in the convolutional layer are local, and the range of extractable features depends on the size of the convolutional kernel. The fully connected layer aggregates the global information from these simplified features into a compact form, maximizing the sharing of all nodes, and enabling the model to capture higher-level relationships among features in the entire input data.

The input data for the fully connected layer is a one-dimensional array of scalar feature values, where each element represents an input feature corresponding to a neuron. It functions as a classifier within the neural network, mapping the learned distributed feature representations to the sample label space for classifying network attacks.

(3) Activation function layer

The activation function is crucial in neural networks, as it introduces nonlinear properties and overcomes the limitation of relying solely on linear feature transformations in feature extraction. This enables

neural networks to learn and represent more complex patterns and relationships, facilitating the capture of nonlinear features in the data.

Selecting the appropriate activation function is essential for the training and effectiveness of neural networks, as various activation functions can address distinct problems. In binary classification problems, the Sigmoid activation function is typically employed to map output values between [0, 1]. In multi-class classification problems, the Softmax activation function is commonly utilized. The Softmax activation function is applied to the output of the fully connected layer to obtain probabilities for multi-class classification, transforming raw data into a probability distribution for the output classes. The Softmax multi-class classifier assigns the probability of each class sample to its corresponding output node, and the generation process is illustrated in the following equation:

$$S_i = \frac{\exp\left(\omega_c^T Z_i\right)}{\sum_{j=1}^{N} \exp\left(\omega_j^T Z_i\right)} \tag{8}$$

Among them, S_i is the output of the classifier, $\omega = \left\{\omega_1^T, \omega_2^T, \cdots, \omega_N^T\right\}$. T is the trainable parameters for the classifier, ω_c is parameters for Class C samples, Z_i is output after bidirectional feature fusion, $c \in \{1, 2, \cdots, N\}$ indicate the category to which it belongs, N is the total number of categories.

This model employs the Softmax activation function to classify input data into five categories: Normal, DoS, Probe, U2R, and R2L attack types. The output layer consists of five neurons, each representing the probability of the input belonging to a specific class, with the class exhibiting the highest probability being identified as the predicted class.

4 Experiment and Result Analysis

In this section, we provide a comprehensive evaluation of the model based on performance indicators and the configuration of hyperparameters for model optimization. To evaluate the model proposed in this article, experiments were conducted in an environment equipped with an Intel Core i7-9700 CPU, Radeon RX550 GPU, 32 GB of RAM, and a 64-bit Windows 11 operating system. Simulation experiments were performed using PyTorch 2.0 and Python 3.9.

4.1 Experimental Dataset

The KDD-CUP99 dataset, widely used as a benchmark in intrusion detection but suffers from various data quality issues, including missing values, inability to directly process discrete features, class imbalance, excessive features, and inconsistent feature scales. These issues can result in significant errors in evaluation results. Such problems hinder researchers from accurately evaluating model performance in experiments and limit the dataset's applicability. The NSL-KDD dataset improves upon by removing redundant and duplicate records, simplifying feature structures, and reasonably adjusting data distribution. These improvements make the design of the training and testing sets more balanced, establishing the dataset as a more effective benchmark for intrusion detection methods.

This study uses the NSL-KDD dataset for experiments, which comprises 125,937 records. The dataset size is moderate, making it suitable for model training and evaluation. The experiment employs a 10-fold cross-validation method. The dataset is divided into 10 equal parts, with 70% used for model training and 30% for testing. This segmentation method ensures a rational distribution of training and testing data while reducing the risk of overfitting.

4.2 Evaluation Metrics

To ensure transparency, readability, and an intuitive understanding of the model's performance, we include the definitions and formulas of key evaluation metrics: accuracy, precision, recall, and F1 score. This not only contextualizes the improvements introduced by our model but also serves as a reference for readers from various backgrounds. The calculation methods for these metrics are detailed below:

Accuracy (Acc) is defined as the percentage of correctly identified data out of the entire dataset:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

Precision (Pre) refers to the proportion of data with positive predicted results that are actually positive:

$$Pre = \frac{TP}{TP + FP} \tag{10}$$

Recall denotes the proportion of genuine positive data among all the data that has been predicted as positive:

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

F1 is a weighted average of precision and recall:

$$F1 = \frac{2TP}{2TP + FN + FP} \tag{12}$$

TP (True Positive) signifies the count of normal data points correctly identified by the model; *TN* (True Negative) represents the number of abnormal data points that are accurately classified; *FP* (False Positive) indicates instances where normal data is wrongly labeled as abnormal; *FN* (False Negative) corresponds to abnormal data points that are incorrectly classified as normal.

4.3 Experimental Parameter Settings

The experiments described in this article were performed in a consistent hardware and software environment. The processed dataset was split into a training set and a test set, maintaining a ratio of 7:3. To prevent overfitting, the dataset is proportionally divided into distinct subsets for each experiment. In this study, the model comprises five network layers, with a dropout rate of 0.2, a learning rate of 0.001, and a total of 100 iterations. The experimental results are illustrated in Figs. 4 and 5, with Fig. 4 depicting accuracy convergence and Fig. 5 depicting loss rate convergence. The blue line indicates convergence in the training set, while the red line indicates convergence in the validation set.

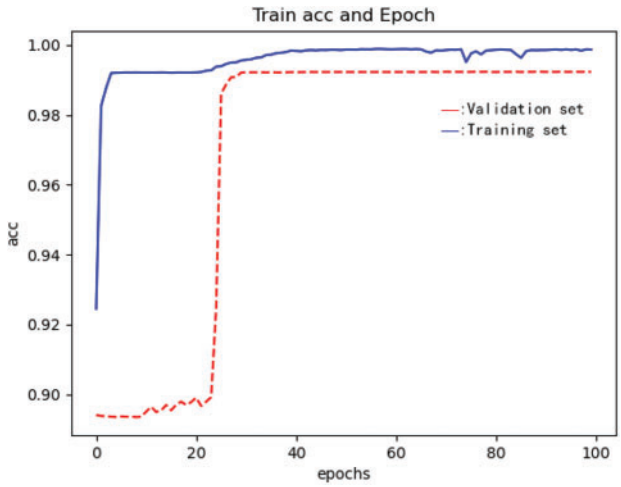

Figure 4: Convergence of model accuracy

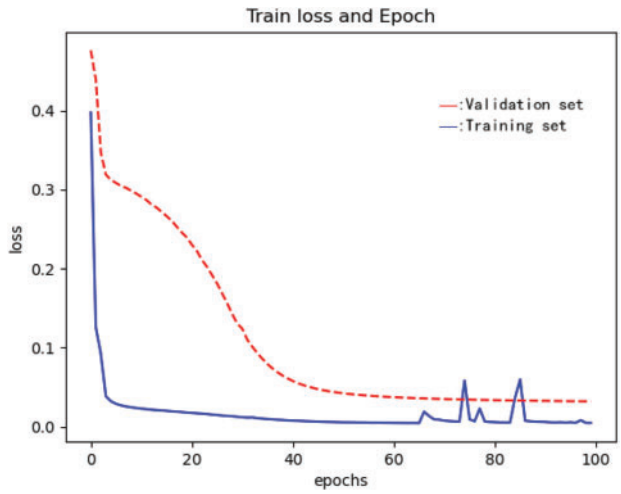

Figure 5: Convergence of model loss rate

4.4 Hyperparameter Optimization

This article uses search-based hyperparameter optimization methods to arrange and combine optional hyperparameters, to conduct experimental testing to obtain the best-performing hyperparameters. The article mainly lists the selected activation functions, batch sizes, and optimizers.

Activation functions: Common activation functions encompass Sigmoid function, ReLU function, Tanh function, and Softmax function. This article uses three of the most commonly used activation functions for comparative experiments, namely Sigmoid function, ReLU function, and Softmax function.

Batch size: The number of samples selected for one training session is called batch size, and different batch sizes can affect the convergence speed and effectiveness of the model. In this experiment, 32 and 64 were used as batch sizes for hyperparameter search.

Optimizer: Common optimizers include stochastic gradient descent (SGD), Adagrad, Momentum, Adam, RMSProp, etc. Different optimizers may have differences in the way and speed of updating parameters, so this article uses three of the most commonly used optimizers for comparative experiments, namely stochastic gradient descent (SGD), Adagrad, and Adam.

This article conducts comparative experiments by selecting different activation functions, optimizers, and batch sizes to achieve optimal performance. Table 2 demonstrates the performance for each combination of hyperparameters. The experimental results show that when the activation function is ReLU, the optimizer is Adam, and the batch size is 64, the accuracy, precision, and F1 score of the hyperparameter selection in this group are the highest, but the recall score is 0.24% lower than the combination of Softmax, Adam, and batch size of 32. Considering the need to optimize the overall performance of the model, this paper chooses ReLU as the activation function, Adam as the optimizer, and a hyperparameter combination with a batch size of 64.

Table 2: The influence of different hyperparameters on model accuracy

Activation function	Optimization	Batch size	Accuracy	Precision	Recall	F1
ReLu	SGD	32	99.24	99.21	99.75	99.48
ReLu	SGD	64	97.64	99.56	98.56	99.06
ReLu	Adam	32	99.71	99.88	99.05	99.46
ReLu	Adam	64	99.86	99.94	99.61	99.82
ReLu	Adagrad	32	99.24	99.08	98.37	98.72
ReLu	Adagrad	64	99.55	99.46	98.91	99.18
Sigmoid	SGD	32	99.42	98.76	97.91	98.83
Sigmoid	SGD	64	99.38	98.97	98.57	98.77
Sigmoid	Adam	32	99.67	99.08	99.60	99.34
Sigmoid	Adam	64	99.73	99.71	99.7	99.70
Sigmoid	Adagrad	32	97.60	97.56	95.62	96.58
Sigmoid	Adagrad	64	97.65	97.99	95.19	96.57
Softmax	SGD	32	99.62	99.37	99.24	99.30
Softmax	SGD	64	99.15	98.17	99.19	98.68
Softmax	Adam	32	99.80	99.56	99.85	99.70
Softmax	Adam	64	99.82	97.57	99.64	99.61
Softmax	Adagrad	32	99.11	97.89	99.69	98.29
Softmax	Adagrad	64	96.18	93.79	99.80	92.27

4.5 Ablation Experiment

To analyze the efficacy of the proposed SA-ResNet model in a quantitative manner, ablation experiments were conducted on the rebalanced dataset. During these experiments, the residual neural network module and the spatial attention mechanism module were systematically removed, while ensuring that the parameters of the replacement model remained consistent.

Fig. 6 presents the results of our ablation study, which systematically evaluates the contribution of each component of the SA-ResNet model. The results demonstrate that the incorporation of our tailored residual blocks and the adapted spatial attention mechanism significantly outperforms the baseline CNN model across all evaluation metrics. Specifically, the recall rate improvement by 0.6% and the substantial

enhancement in F1 score underscore the effectiveness of our proposed enhancements. The CNN+ResNet model outperforms the single CNN model in accuracy, precision, and F1 score. However, CNN+ResNet model has a lower recall score than the single CNN model due to the extensive feature set required for training the residual neural network. The results indicate that the CNN with the spatial attention mechanism (CNN+SA) scored the lowest among the four evaluation metrics. Building upon the CNN+ResNet model, a spatial attention mechanism module is integrated. By concentrating on relevant data features, the spatial attention mechanism enhances the recall rate by 0.3% compared to the CNN+ResNet model and significantly outperforms other models in the remaining three evaluation metrics, thus validating the effectiveness of our model. The proposed SA-ResNet model achieves an accuracy of 0.9986, a recall rate of 0.9961, and an F1 score of 0.9982.

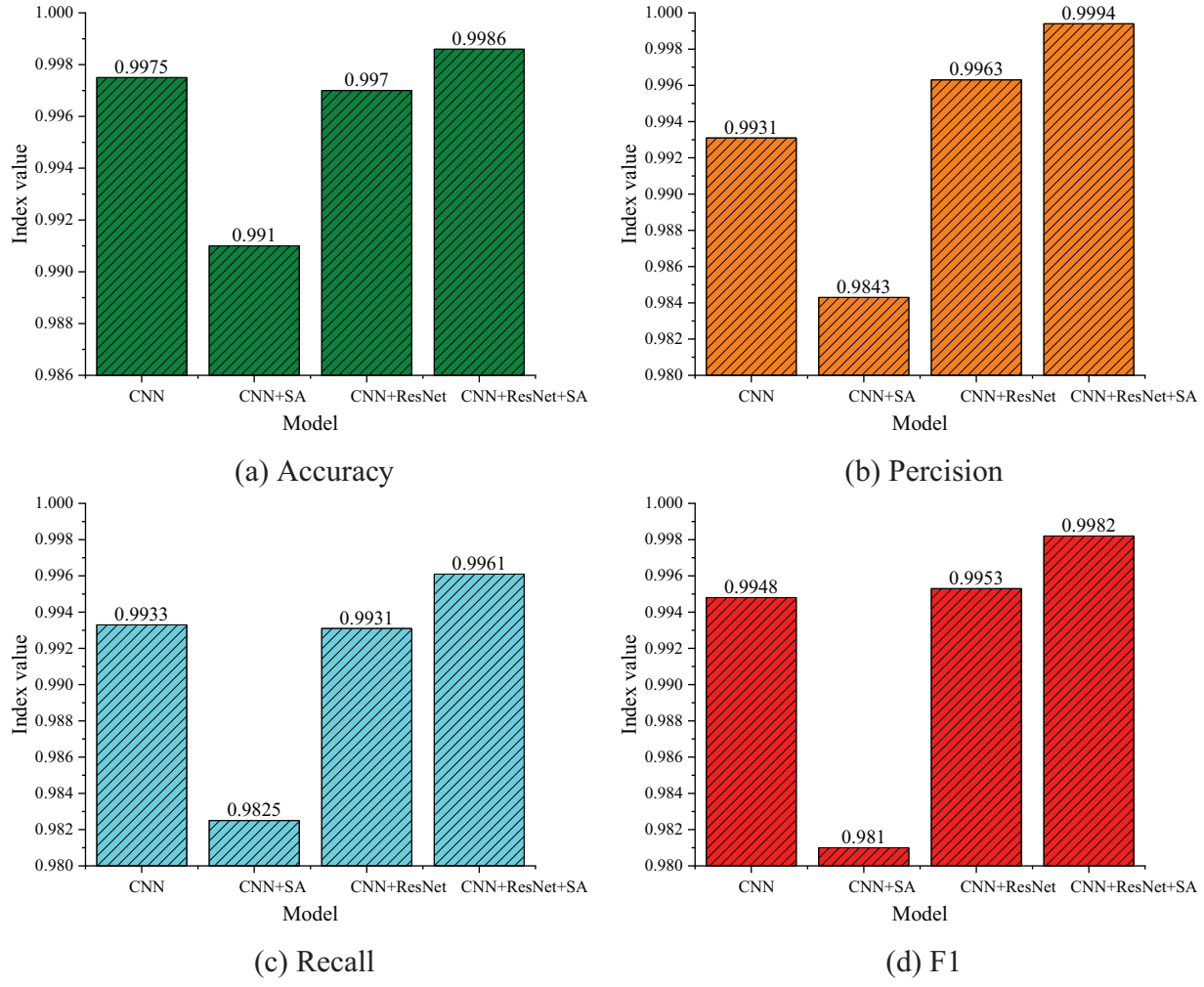

Figure 6: Results of ablation experiment

4.6 Performance Comparison Analysis Experiment

This article compares the performance of the proposed method with six existing deep learning algorithms on the NSL-KDD dataset across various aspects. As shown in Table 3, the accuracy of the proposed model is as high as 99.86%, which is 2.8% higher than that of the Spiral Convolution-LSTM (SCL) Model.

Additionally, the accuracy of the model reaches 99.94%, 2.86% higher than that of the SCL Model. However, compared to Attention-BiTCN, the recall rate is 0.09% lower. Although the model has some shortcomings in recall rate, its use of residual structures and the learning of deeper features lead to better performance in accuracy, precision, and F1 score. This indicates that the model is more effective at avoiding false positives. Compared to existing deep learning models like TCN and Artificial Neural Networks (ANN), this article employs spatial attention mechanism modules to enhance the model's ability to capture key information while suppressing the influence of irrelevant or redundant information. This allows the model to better identify the characteristics of malicious traffic, thereby improving recognition accuracy.

Table 3: Comparison of methods

Method	Accuracy	Percision	Recall	F1
Res-CNN-SRU [7]	98.79	95.34	95.04	95.19
SCL-Model [14]	97.06	97.08	97.06	97.07
BiTCN [16]	99.7	99.69	99.70	99.69
ANN [25]	94.69	99.69	78.92	75.45
TCN [26]	99.35	99.36	99.35	99.35
GRU-RNN [27]	99.13	99.05	99.08	99.06
Ours	99.86	99.94	99.61	99.82

Overall, the feature extraction model proposed in this paper demonstrates greater comprehensiveness and accuracy for network data traffic intrusion detection on this dataset, resulting in strong detection performance and significant application value.

5 Conclusion

To address the low accuracy of network attack detection in existing intrusion detection models, this paper proposes a method that integrates a SA-ResNet model. The primary contribution of this article is the incorporation of a spatial attention mechanism into the residual neural network, allowing the model to dynamically focus on the most relevant areas within the input data. This enhancement improves the model's ability to capture key information, enabling better detection of network intrusion traffic characteristics and overall network performance. The proposed model incorporates residual structures, which effectively mitigate the vanishing gradient problem, prevent overfitting, and enhance the network's convergence speed. Compared to other intrusion detection methods, this paper demonstrates superior model prediction accuracy, providing valuable insights for the field of intrusion detection. While the SA-ResNet model has shown superior performance in accuracy and F1 score, there is room for improvement in recall rate. Future work will focus on further optimizing our spatial attention mechanism to better capture the nuances of network traffic data, Additionally, we plan to explore the generalization ability of our model to different network environments and attack vectors, ensuring its robustness and applicability in diverse scenarios.

Acknowledgement: The authors would like to thank the experimental team for their strong cooperation, the tutor for their full support in the research process, and the editors and reviewers for each piece of valuable advice.

Funding Statement: This research is partially supported by National Natural Science Foundation of China (62473341), Key Research and Development Special Project of Henan Province (221111210500), Key Research and Development Special Project of Henan Province (242102211071, 242102210142, 232102211053).

Author Contributions: Study conception and design: Yuming Dai, Zengyu Cai; data collection: Yuan Feng; analysis and interpretation of results: Yuming Dai, Jianwei Zhang; draft manuscript preparation: Yuming Dai. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available on request from the corresponding author, Jianwei Zhang, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Deore B, Bhosale S. Hybrid optimization enabled robust CNN-LSTM technique for network intrusion detection. IEEE Access. 2022;10:65611–22. doi:10.1109/ACCESS.2022.3183213.
2. Anderson JP. Computer security threat monitoring and surveillance. Technical Report, James P. Anderson Company. 1980 [cited 2024 Jan 4]. Available from: https://cir.nii.ac.jp/crid/1573950399661362176.
3. Denning D. An intrusion-detection model. IEEE Trans Softw Eng. 1987;13(2):222–32. doi:10.1109/TSE.1987.232894.
4. Peng W, Kong X, Peng G, Li X, Wang Z. Network intrusion detection based on deep learning. In: 2019 International Conference on Communications, Information System and Computer Engineering (CISCE); IEEE; 2019. p. 431–5.
5. Akhtar MA, Qadri SMO, Siddiqui MA, Mustafa SMN, Javaid S, Ali SA. Robust genetic machine learning ensemble model for intrusion detection in network traffic. Sci Rep. 2023;13(1):17227–748. doi:10.1038/s41598-023-43816-1.
6. Altaha M. Ae-based network intrusion detection for dnp packet injection attacks. Dbpia. 2020;36:508–13.
7. Cai Z, Si Y, Zhang J, Zhu L, Li P, Feng Y. Industrial internet intrusion detection based on Res-CNN-SRU. Electronics. 2023;12(15):3267–83. doi:10.3390/electronics12153267.
8. Deore B, Bhosale S. Intrusion detection system based on RNN classifier for feature reduction. SN Comput Sci. 2022;3(2):114–22. doi:10.1007/s42979-021-00991-0.
9. Muthunambu NK, Prabakaran S, PrabhuKavin B, Siruvangur KS, Chinnadurai K, Ali J. A novel eccentric intrusion detection model based on recurrent neural networks with leveraging LSTM. Comput Mater Contin. 2024;78(3):3089–127. doi:10.32604/cmc.2023.043172.
10. Li S, Li Q, Li M. A method for network intrusion detection based on GAN-CNN-BILSTM. Int J Adv Comput Sci Appl. 2023;14(5):507–15. doi:10.14569/issn.2156-5570.
11. Lee J-H, Kim J-W, Choi M-J. SSAE-deepcnn model for network intrusion detection. In: 2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS); IEEE; 2021. p. 78–83.
12. Zhao R, Wang Y, Xue Z, Ohtsuki T, Adebisi B, Gui G. Semisupervised federated-learning-based intrusion detection method for internet of things. IEEE Internet Things J. 2023;10(10):8645–57. doi:10.1109/JIOT.2022.3175918.
13. Yang XW, Zhang J, Kuang LQ, Pang M. Network intrusion detection model integrating CNN-BIGRU and attention mechanism. Inform Secur Res. 2024;10(3):202–8.
14. Wang F, Dong Z. Fusion of spiral convolution-LSTM for intrusion detection modeling. Comput Mater Contin. 2024;79(2):2315–29. doi:10.32604/cmc.2024.048443.
15. Alrayes FS, Zakariah M, Amin SU, Khan ZI, Alqurni JS. CNN channel attention intrusion detection system using NSL-KDD dataset. Comput Mater Contin. 2024;79(3):4319–47. doi:10.32604/cmc.2024.050586.
16. Sun HZ, Wang J, Wang P, An YL. Network intrusion detection method based on attention-BITCN. Inform Netw Secur. 2024;24(2):309–18.
17. Wu Z, Zhang H, Wang P, Sun Z. RTIDS: a robust transformer-based approach for intrusion detection system. IEEE Access. 2022;10(3):64375–87. doi:10.1109/ACCESS.2022.3182333.
18. Ullah S, Ahmad J, Khan MA, Alshehri MS, Boulila W, Koubaa A, et al. TNN-IDS: transformer neural network-based intrusion detection system for MQTT-enabled IoT networks. Comput Netw. 2023;237(5):110072. doi:10.1016/j.comnet.2023.110072.

19. Friji H, Olivereau A, Sarkiss M. Efficient network representation for GNN-based intrusion detection. In: Tibouchi M, Wang X, editors. Applied cryptography and network security. Cham: Springer Nature Switzerland; 2023. p. 532–54.
20. Seo E, Song HM, Kim HK. Gids: gan based intrusion detection system for in-vehicle network. In: 2018 16th Annual Conference on Privacy, Security and Trust (PST); IEEE; 2018. p. 1–6.
21. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); IEEE; 2016 Jun.
22. Ma Z, Li X. An improved supervised and attention mechanism-based U-Net algorithm for retinal vessel segmentation. Comput Biol Med. 2024;168(2):107770. doi:10.1016/j.compbiomed.2023.107770.
23. Laghrissi F, Douzi S, Douzi K, Hssina B. IDS-attention: an efficient algorithm for intrusion detection systems using attention mechanism. J Big Data. 2021;8(1):149. doi:10.1186/s40537-021-00544-5.
24. Xu R, Zhang Q, Zhang Y. TSSAN: time-space separable attention network for intrusion detection. IEEE Access. 2024;12(2):98734–49. doi:10.1109/ACCESS.2024.3429420.
25. Zakariah M, AlQahtani SA, Alawwad AM, Alotaibi AA. Intrusion detection system with customized machine learning techniques for NSL-KDD dataset. Comput Mater Contin. 2023;77(3):4025–54. doi:10.32604/cmc.2023.043752.
26. Liu JZ, Liu S, Zhang J. An industrial intrusion detection method based on hybrid convolutional neural networks with improved TCN. Comput Mater Contin. 2024;78(1):411–33. doi:10.32604/cmc.2023.046237.
27. Li J, Xia S, Lan H, Li S, Sun J. Network intrusion detection method based on GRU-RNN. J Harbin Eng Univ. 2021;42(6):879–84.