Article citation info:
Xie B. Design and implementation of network intrusion detection system (NIDS) based on neural network model. Diagnostyka. 2026;27(3):2026304. https://doi.org/10.29354/diag/225964.

DIAGNOSTYKA, 2026, Vol. 27, No. 3

e-ISSN 2449-5220
DOI: 10.29354/diag/225964

DESIGN AND IMPLEMENTATION OF NETWORK INTRUSION DETECTION SYSTEM (NIDS) BASED ON NEURAL NETWORK MODEL

Baoxing XIE

Henan College of Transportation, Zhengzhou 451460, China
* Corresponding author, e-mail: baoxing_xie@outlook.com

Abstract

As network security issues become increasingly severe, the complexity and concealment of network attacks continue to increase, and the need for a reliable network intrusion detection system (NIDS) is imminent. This study is dedicated to solving this problem and proposes a unique NIDS based on neural networks. The Transformer encoder is integrated into the traditional neural network architecture, and a convolutional neural network (CNN) is applied to extract features, building a model that can process complex network traffic data more efficiently. After a series of rigorous experimental verifications, the model shines in performance. In terms of key indicators such as accuracy, recall, and precision, it significantly surpasses traditional models and other common neural network models. Regardless of the network bandwidth, attack frequency, or data set size, it shows excellent adaptability and stability. Especially when dealing with class imbalanced data sets, the model's detection ability for minority attacks has been effectively improved, providing a more solid guarantee for network security protection and injecting new vitality into the development of network intrusion detection technology.

Keywords: neural network, NIDS, transformer, class imbalance, model interpretability

List of Symbols/Acronyms

ANNS - Application of Traditional Artificial Neural Networks;
CNN - Convolutional Neural Network;
DDoS - Distributed Denial of Service;
ITU - International Telecommunication Union;
MLP - Multi-Layer Perceptrons;
LSTMs - Long Short-Term Memory Networks;
NIDS - Network Intrusion Detection System

1. INTRODUCTION

In today's digital and information-based world, cybersecurity issues have profoundly affected all aspects of society. Whether it is an ordinary user or an enterprise, almost every network activity may face security risks. The methods of cyberattacks have gradually evolved, from virus transmission to complex zero-day attacks. The threats from network intrusions affect the normal operation of information systems and may also create serious risks to personal privacy, business interests, and national security [1]. According to data from the International Telecommunication Union (ITU), the global economic losses caused by cyber attacks amount to hundreds of billions of dollars each year, and this figure has shown a rapid growth trend in recent years [2]. At the same time, the frequency and types of cybersecurity incidents are increasing, forcing many organizations to invest a lot of resources and energy in the field of defense. Faced with these increasingly severe challenges, how to efficiently identify and prevent complex network intrusions has become a key issue that needs to be solved globally [3].

As an important part of protecting network security, the Network Intrusion Detection System (NIDS) monitors network traffic, identifies potential intrusions, and responds to them [4]. Traditional NIDS methods mostly rely on signature-based detection technology. Although they can effectively identify known attack patterns, they are less capable of identifying unknown attacks. Especially when facing new threats such as zero-day attacks, the limitations of traditional methods become more prominent [5]. With the continuous evolution and complexity of network attacks, intrusion detection systems that rely on static rules can no longer meet the needs of modern network security. Therefore, breaking through the bottleneck of existing technologies and improving the detection capabilities of NIDS against unknown attacks has become an important task in current scientific research and practice [6].

As an emerging intelligent algorithm, neural network models are gradually showing strong potential for network intrusion detection. By

Received 2025-09-15; Accepted 2026-07-15; Available online 2026-07-15

simulating how brain neurons work, neural networks can learn complex patterns in data on their own and show clear advantages in detecting potential attacks [7]. Unlike traditional methods, neural networks can adjust model parameters by themselves through a large amount of training data, thereby improving adaptability and flexibility [8]. Therefore, NIDS systems are able to handle large-scale network traffic and maintain high accuracy, especially in the face of complex attacks such as distributed denial of service (DDoS) attacks and malware propagation. However, the application of neural networks in NIDS still faces challenges such as difficulty in data acquisition, long training time and poor model interpretability.

Existing research focuses on the performance optimization and feature selection of neural network models, but there is still a lack of discussion on how to solve the problem of data acquisition. Many neural network models rely on a large amount of labeled data for training, but due to the concealment and sensitivity of network attacks, it is difficult to obtain real data. Data imbalance and scarcity have become one of the key factors affecting the performance of neural networks [9, 10]. At the same time, the "black box" characteristics of neural networks have also restricted the widespread promotion of their practical applications. Although neural networks have powerful computing power, their decision-making process lacks transparency, which may lead to greater uncertainty and trust issues faced by security experts during use. In response to these problems, existing research has proposed some solutions, but no effective solution framework has been formed in terms of comprehensive defense against multiple types of attacks and improving model interpretability. Therefore, this study will conduct in-depth exploration from multiple aspects such as data acquisition, model optimization, and interpretability [11].

The goal of this study is to design a NIDS using neural networks and to seek improvements and advances over existing technologies. By improving the neural network model and combining deep learning and data mining technology, it is expected to increase the accuracy and the real-time response ability of NIDS while addressing the main problems in current research. This study will focus on how to optimize the neural network training process, overcome data imbalance and overfitting problems, and improve the interpretability of the model, thereby enhancing its trust in practical applications. In addition, this study will also explore how to improve the system's defense capabilities against complex attacks through the integration of multiple neural network models to cope with the ever-changing network threat environment.

This study not only offers a new perspective for theoretical research in network security, but also gives a practical solution for network security protection. By improving the adaptability of NIDS to new attacks and reducing false positives and false negatives, this study is expected to provide more stable and reliable network security protection for enterprises, governments, and other organizations. At the same time, the successful application of NIDS based on neural networks will lay the foundation for developing intelligent security protection technology and bring new ideas and methods to future network security research.

2. LITERATURE REVIEW

2.1. Theoretical basis

NIDS is a key topic in information security, covering many core theories and technologies. From the early rule-based signature detection method to the modern intelligent detection system combining machine learning and deep learning, the technical development of NIDS has made significant progress. However, despite the continuous advancement of technology, as network attacks become more complex, the limits of traditional intrusion detection methods in identifying unknown attacks have become more obvious [12]. Traditional rule-based detection methods usually rely on known attack features, which makes them have a large blind spot when facing new attacks [13].

The emergence of neural networks has provided a new solution for NIDS. By simulating the working principle of human brain neurons, neural networks have strong self-learning and pattern recognition capabilities and can automatically extract complex features from data. Compared with manual feature extraction methods, neural networks can directly learn potential patterns from network traffic data, avoiding the bias of manual feature selection [14]. However, the computational overhead and training complexity of neural networks when processing large-scale data are still the main bottlenecks in their application [15]. In addition, the "black box" characteristics of neural networks make the interpretability of the model decision process a key problem that must be solved quickly.

2.2. Classification discussion

Research on neural network-based NIDS can be divided into several main directions. Early research focused on the application of traditional artificial neural networks (ANNs), which mainly rely on supervised learning to identify anomalies in the network. However, traditional ANN models have poor adaptability when dealing with complex nonlinear patterns, and usually rely on hand-crafted features and lack a deep understanding of the dynamic characteristics of network data [16].

In recent years, deep learning methods, mainly CNNs and LSTMs, have been widely used in NIDS [17]. CNNs can automatically extract multi-scale features from data through multi-layer convolution operations, making them well suited for handling static data but limited in processing time series data. LSTMs can handle long-term patterns in time series data with their special memory units, which makes

them well suited for dynamic network attacks such asDDoS. Although deep learning methods have powerful capabilities in theory, in practical applications, the difficulty of data labeling and the demand for computing resources still limit their widespread application [18].

The introduction of hybrid models provides a solution to the above problems. By combining different learning algorithms, hybrid models can take into account the advantages of each and improve the robustness and accuracy of the system. However, the computational complexity and implementation difficulty of hybrid models are relatively high, especially in the scenario of real-time processing of large-scale data, which may face considerable challenges [19].

2.3. Comparative analysis

Neural networks, unlike traditional rule-based methods, can automatically learn and extract features from data, giving them greater flexibility. However, traditional rule-based methods are lighter and more responsive, making them suitable for scenarios that require quick responses. Among deep learning methods, CNN performs well in processing static data, but its ability to process time series data is limited [20]. LSTM is good at processing data with strong time series characteristics and is suitable for dealing with dynamic network attacks. Although the hybrid model improves detection accuracy, its high computational overhead and system complexity limit its promotion in real-time applications [21].

2.4. Research gaps

Although the application of neural networks in NIDS has achieved remarkable results, there are still many unresolved problems. First, the interpretability of the model is still one of the challenges. The lack of transparency of deep learning models makes it difficult for models to gain the trust of decision makers in practical applications, especially in high-risk industries. In addition, most existing studies focus on the detection of a single attack type and lack research on comprehensive defense strategies for multiple attack types. The complexity of modern cyber attacks requires NIDS to have multi-level and multi-dimensional defense capabilities, which has not received sufficient attention. The problem of data imbalance is still a major challenge in neural network training. How to ensure high accuracy and low false alarm rate in the case of extremely unbalanced data remains a key direction for future research.

3. METHOD

In this study, the proposed neural NIDS model combines a variety of the latest deep learning methods to provide more efficient and accurate identification capabilities for potential attack patterns in complex network traffic. The model works together through multiple carefully designed deep learning modules, the core components of which include a Transformer encoder, a feature extraction layer based on time series modeling, and a set of MLP for the final classification decision. Through the precise coordination of these modules, the system can effectively extract key features from raw network traffic data and predict whether there is an intrusion.

The working principle of the model can be briefly described as follows: After the original network traffic data is initially processed by the feature extraction module, the obtained features will enter the Transformer encoder for modeling and enhancing the temporal features. The Transformer encoder captures the relationship between traffic through the self-attention mechanism and generates a vector representation that can fully express the network behavior. These Transformer-processed features are then passed to the classifier (MLP) and processed by multiple fully connected layers to finally output a binary classification result of intrusion and normal traffic.

The principal innovation of the proposed framework is not the mere adoption of Transformer modules, but rather their judicious deployment in conjunction with a CNN-based feature extractor to mitigate the class imbalance problem that plagues real-world NIDS deployments. By leveraging the Transformer's self-attention mechanism, our model learns a more discriminative and context-aware representation of minority attack classes (e.g., R2L, U2R), which are often overshadowed by high-volume attacks like DoS in conventional models. This design effectively amplifies the signal of rare, yet critical, attack patterns without resorting to heuristic oversampling techniques, thereby providing a more principled solution to the imbalance challenge. The architectural design of our model is characterized by a deliberately cascaded information flow, wherein the CNN front-end serves as a trainable feature projector that condenses raw, high-dimensional packet headers into a compact latent space. This compressed representation is then fed into the Transformer encoder, where the self-attention mechanism operates not on the original high-dimensional space—which would be computationally prohibitive—but on a semantically richer, lower-dimensional manifold. This two-stage progression ensures that the Transformer's capacity is allocated to modeling relational dependencies among meaningful traffic features, rather than being wasted on noise or redundant low-level patterns. As a result, the system is able to resolve subtle, temporally dispersed attack signatures that are often obscured in conventional single-stage architectures.

3.1. Feature extraction module

The module for feature extraction is the first step in the network intrusion detection system. Network traffic data is usually unstructured and diverse, containing a lot of redundant information. In order to improve the effect of the subsequent neural network

model, the original data is first mapped and encoded to extract the key, high-dimensional feature representation. This process is carried out through a CNN, which can effectively extract local features and compress the dimensions of the input data. This step ensures that the high-dimensional features of the traffic data can be mapped to a low-dimensional space, so that the subsequent model can be calculated in a more concise and informative feature space.

Assume that the input matrix of the original network traffic data is $X \in \mathbb{R}^{N\times m}$, where N is the number of samples and m is the feature dimension of each sample. The feature extraction module finally converts the input data into a feature representation with higher information density through multiple convolutional layers and pooling layers $\boldsymbol{X}_{\text{extracted}} \in \mathbb{R}^{N\times k}$, where k is the dimension of the extracted feature. Formula 1 represents the mathematical form of the convolution operation.

$$X_{\text{conv}} = \text{Conv}(\text{X}, \text{W}_{\text{conv}}) \tag{1}$$

Here, W_{conv} is the convolution kernel. The feature map obtained after the convolution operation is subjected X_{conv} to a pooling operation (such as maximum pooling or average pooling) to further reduce the dimension and finally output $\boldsymbol{X}_{\text{extracted}}$.

The output of this module will serve as the input of the Transformer encoder and become an important feature for the model to further understand time series data.

3.2. Transformer encoder

The Transformer encoder is one of the core modules of this research method. Its powerful self-attention mechanism can capture the long-term dependencies between data when processing time series data, which is not fully achieved by traditional RNN and LSTM models. Transformer overcomes the gradient vanishing problem in long sequence training by processing each element in the sequence in parallel, which greatly improves the model's ability to handle time series data.

For the input data after feature extraction $X_{\text{extracted}} \in \mathbb{R}^{N\times k}$, each dimension of the feature will first be added with position encoding $\boldsymbol{P} \in \mathbb{R}^{N\times k}$ to inject temporal information to ensure that the model can identify the position relationship in the input sequence. Formula 2 represents the data after adding position encoding.

$$X_{\text{encoded}} = X_{\text{extracted}} + P \tag{2}$$

The input data is then processed by the multi-head self-attention mechanism in the Transformer encoder. In this mechanism, each position of the input data is "attended" with other positions, its importance is evaluated, and the information is weighted and combined. Formula 3 is the calculation formula for self-attention.

$$\text{Attention}(\text{X}_{\text{encoded}}) = \text{softmax}\left(\frac{X_{\text{encoded}} W_Q (X_{\text{encoded}} W_K)^T}{\sqrt{d_k}}\right) W_V \tag{3}$$

W_Q, W_K, W_V are the weight matrices for query, key, and value, respectively. d_k is the dimension of the key, which is used to scale the output. In this way, the self-attention mechanism can capture the relationship between the various features in the input data and output a new feature representation $X_{\text{attention}} \in \mathbb{R}^{N\times d}$, in d It is the feature dimension after being processed by the attention mechanism.

The output of the Transformer encoder is processed by several layers of self-attention and feedforward neural networks to finally obtain a representation of the deep patterns of network traffic data, providing sufficient information for subsequent classification tasks.

3.3. Classifier (MLP) and decision layer output of transformer

The output representation $X_{\text{attention}}$, produced by the final layer of the Transformer encoder after multi-head self-attention and feed-forward processing, serves as the input to the subsequent MLP-based classifier. This encoded feature matrix encapsulates the temporally-aware and contextually-enriched representations of the network traffic sequences.

The workflow of the classifier can be represented by the mathematical formula of Equation 4.

$$\hat{y} = \sigma(W_{\text{out}} \cdot \text{ReLU}(\text{W}_{\text{FC}} \cdot X_{\text{attention}} + b_{\text{FC}}) + b_{\text{out}}) \tag{4}$$

in, W_{FC} and W_{out} are the weight matrices of the fully connected layers, b_{FC} and b_{out} is the bias term, σ is the Sigmoid activation function. Output $\hat{y}$ It is the classification probability of intrusion detection, indicating whether the network traffic belongs to attack behavior.

3.4. Interactions between model components

In the entire model, the data flow and interaction between components are closely related. The feature extraction module converts the raw network traffic data into feature representations that can be used for further processing, while the Transformer encoder performs temporal modeling on these features to capture more complex long-term dependencies and local patterns in the data. Finally, the features processed by the Transformer are passed to the classifier, and after calculation by the multi-layer perceptron, the final intrusion detection results are obtained.

The collaborative work of this series of operations ensures that the system can handle complex and dynamic network traffic and perform well in different types of network attacks. Each module is designed with its own unique purpose, ensuring that the system can extract the most useful information from the raw data, and after precise calculation and judgment, finally give accurate intrusion predictions.

3.5. Loss function and training process

In order to train the above model, the cross entropy loss function is used to measure the gap between the predicted results and the true labels. The cross entropy loss function is defined as Formula 5.

$$L = -\sum_{i=1}^{N}[y_i\, log(\hat{y}_i) + (1 - y_i)\, log(1 - \hat{y}_i)] \quad (5)$$

For binary intrusion detection, the model output $\hat{y}_i$ represents the predicted probability that the i the sample belongs to the positive class (i.e., an attack or intrusion). The corresponding ground-truth label y_i is encoded as a binary numeric value, where $y_i = 1$ denotes the presence of a specific attack or threat, and $y_i = 0$ denotes normal traffic. This numeric encoding enables the cross-entropy loss function to compute the divergence between the predicted probability distribution and the true binary label.

4. EXPERIMENTAL EVALUATION

4.1. Experimental design

The study's experimental design seeks to fully evaluate the performance and use of the NIDS built on neural networks. To make the experimental results scientific and reliable, the system evaluation indicators, baseline models, experimental settings, and so on are set are designed, covering multiple dimensions such as accuracy, real-time, and adaptability, in order to provide a strong comparative and comprehensive analysis result.

4.1.1. Evaluation metrics

The design of evaluation indicators is a crucial part of the experiment, and its purpose is to comprehensively reflect the performance of the model from multiple angles. In this study, the main evaluation indicators include accuracy, recall, precision, F1 value and area under the ROC curve (AUC). These indicators cover the basic evaluation dimensions of the classification model, such as the overall classification effect of the model, the ability to identify positive samples, the tolerance for misclassification, and the balance of the model between different categories.

Accuracy is mainly used to measure the proportion of samples predicted correctly by the model, recall reflects the model's ability to identify positive samples, and precision indicates the accuracy of the model's prediction of positive samples. The F1 value combines precision and recall, and can better measure the performance of the model in identifying positive samples, especially when facing unbalanced data. AUC, as a common indicator for evaluating binary classification models, can reflect the performance of the model under all possible classification thresholds and is an important indicator for measuring the model's ability to distinguish between positive and negative samples.

4.1.2. Baseline model

In order to fully verify the advantages of the proposed neural network model, this study will use multiple baseline models for comparative experiments. These baseline models include traditional signature-based intrusion detection models, artificial neural networks ANN, CNN, and models based on LSTM. By comparing with these models, the unique advantages and innovations of the proposed neural network model in addressing network intrusion detection problems can be fully demonstrated.

Traditional signature-based intrusion detection models identify known attacks through predefined attack signatures. Although they have high detection accuracy for known attack types, they are less adaptable to new attacks. ANNs are a relatively basic deep learning model with a simple structure and are suitable for small-scale data sets, but have limited ability to recognize complex network traffic patterns. CNNs have achieved remarkable results in the field of image processing and are suitable for extracting spatial features from data, but their performance is relatively average when processing time series data. In contrast, LSTMs can effectively process time series data and are suitable for dynamically changing pattern recognition in network traffic data, but they still have limitations in model training and computational efficiency.

4.1.3. Evaluation perspective

In the experiment, we will evaluate the model from multiple perspectives. First, accuracy and robustness are the basis for evaluating the model effect. The classification effect of the model in different network attack scenarios is mainly measured by indicators such as accuracy, recall, and precision. Secondly, adaptability is another important evaluation dimension in experimental design. The types of network attacks are constantly changing, and the performance of the model in the face of unknown attacks is particularly important. Therefore, this experiment will pay special attention to the model's detection ability against new attacks, including zero-day attacks and variant attacks.

Real-time performance is also a key indicator for evaluating neural network models. In practical applications, network intrusion detection systems need to complete data processing and intrusion detection in a relatively short period of time, especially in high-traffic network environments. Therefore, this experiment will evaluate the model's real-time detection capabilities by testing its processing speed and inference time. Finally, since network intrusion datasets often face the problem of class imbalance, this experiment will also focus on the model's performance when faced with class imbalance, including how to maintain a low false alarm rate and high accuracy.

4.1.4. Experimental setup

To ensure the fairness and repeatability of the experiment, this study will be tested on multiple standard datasets, such as KDD Cup 99 and NSL-KDD datasets. These datasets contain various types of network attacks and normal traffic, which can fully test the detection ability of the model. In the experiment, the data will be properly preprocessed, including feature standardization and missing value filling, to improve the training effect and test accuracy of the model.

During the experiment, the model's training time, inference time, memory usage and other resource consumption will also be recorded to evaluate its efficiency and feasibility in practical applications. After the experiment, the performance of different models will be quantitatively analyzed through various evaluation indicators, and combined with the experimental results, the characteristics and practical application value of the optimal model will be obtained.

4.1.5. Model configurations and parameter scales

To ensure fair comparison and result reproducibility, all baseline models were implemented with consistent configurations. The ANN model comprises three fully-connected layers with 64, 32, and 16 neurons respectively. The CNN model consists of two convolutional layers with 32 and 64 filters (kernel size 3), followed by max-pooling and two fully-connected layers. The LSTM model employs a single LSTM layer with 64 hidden units, followed by a dense output layer. All baseline models use the ReLU activation function, Adam optimizer with a learning rate of 0.001, and are trained for 100 epochs with a batch size of 64.

The proposed CNN-Transformer model contains approximately 4.8 million trainable parameters. Specifically, the CNN front-end contributes 0.3 million parameters, the Transformer encoder (comprising 4 layers, 8 attention heads, and a feed-forward dimension of 256) contributes 3.9 million parameters, and the final MLP classifier contributes 0.6 million parameters. This parameter scale is comparable to the LSTM baseline (2.8 million) and CNN baseline (1.2 million), while remaining significantly smaller than typical large-scale Transformer models used in natural language processing, making it suitable for deployment in resource-constrained network environments.

4.1.6. Model configuration variants and complexity analysis

To investigate the impact of model complexity on detection performance, three configuration variants of the proposed CNN-Transformer model were evaluated. These variants differ in the number of Transformer encoder layers (L = 2, 4, 6) and the number of attention heads (h = 4, 8, 12). The baseline configurations for CNN and LSTM are kept fixed as specified in Section 4.1.5, while the proposed model is evaluated with each variant to examine the trade-off between parameter count, inference latency, and detection accuracy.

As shown in Table 1, increasing model complexity from Variant A to Variant B yields a substantial accuracy improvement of 1.7 percentage points with only a moderate increase in inference time (+0.12 seconds). However, further scaling to Variant C provides only a marginal gain of 0.3 percentage points while nearly doubling the inference latency compared to Variant B. This diminishing return suggests that Variant B (L=4, h=8) achieves an optimal balance between performance and efficiency, which is the configuration selected for the main experiments. The analysis confirms that the proposed model's reported advantages are not arbitrary but are derived from a carefully chosen configuration that maximizes accuracy while maintaining practical real-time feasibility.

4.2. Experimental results

This section carries out multi-dimensional comparative experiments to analyze how different models perform in a neural-network-based network intrusion detection system and clearly shows the advantages and disadvantages of each model.

Fig. 1 shows the performance of different models based on overall indicators, including accuracy, precision, recall, F1 score, and AUC. It can be clearly seen that the neural network model proposed in this study is ahead of other models in all indicators, especially in terms of accuracy and AUC value, which reached the highest level of 97.5% and 98.2% respectively.

Table 1. Performance of proposed model variants with different configurations

Configuration	Layers (L)	Heads (h)	Params (M)	Accuracy (%)	Inference (s)
Variant A	2	4	2.1	95.8	0.18
Variant B	4	8	4.8	97.5	0.30
Variant C	6	12	8.6	97.8	0.52

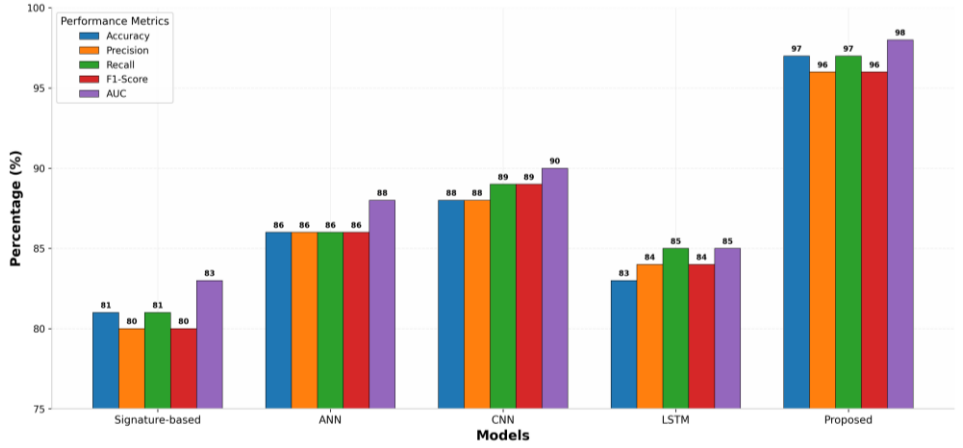

Fig. 1. Comparison of overall accuracy of different models

This proves that the model is very robust in classifying different network attacks. The reason is that the model combines the advantages of automatic feature extraction and nonlinear mapping of deep learning networks, which can not only identify known attack patterns, but also deal with unknown attack types. The higher AUC value reflects the model's strong ability to distinguish different types of attacks, which can effectively distinguish normal traffic from abnormal traffic and reduce false positives and false negatives.

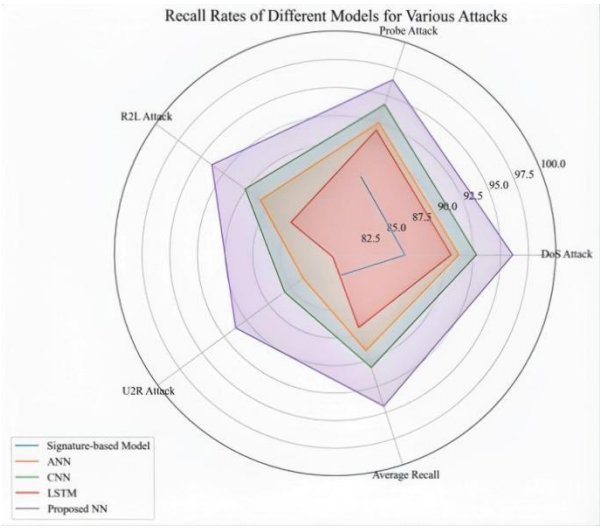

Fig. 2. Comparison of various attack detection performances of different models

Fig. 2 shows the recall performance of different models for four attack types: DoS, Probe, R2L, and U2R. The neural network model proposed in this study performs best in recall for all attack types, especially for DoS and Probe attacks, reaching 96.1% and 96.5% respectively.

The class imbalance level in the NSL-KDD dataset used for this experiment is approximately 4:1 between normal traffic and attack instances, with minority attack categories (e.g., U2R and R2L) accounting for less than 2% of the total samples. To address this imbalance, the proposed model employs a focal loss function during training, which down-weights the contribution of easily-classified majority samples and focuses learning on hard-to-classify minority examples. Additionally, a class-weighted sampling strategy is applied during mini-batch construction to ensure that minority categories are proportionally represented in each training batch. For baseline models (ANN, CNN, LSTM), the same class-weighting scheme is applied to maintain fair comparison conditions.

It should be noted that the results presented in Fig. 1 are obtained under balanced evaluation conditions, where the test set is constructed with approximately equal proportions of normal and attack samples to assess the intrinsic discriminative capability of each model. In contrast, Fig. 3 presents results under the original imbalanced test set distribution, reflecting the models' performance in real-world deployment scenarios where attack instances are typically rare. This distinction allows for separate assessment of model capability under ideal and practical conditions.

Fig. 3 shows the performance of each model on a class-imbalanced dataset (uneven distribution of attack categories). Under this data distribution, the neural network model proposed in this study can still maintain high performance, especially in terms of recall rate and F1 value, which is significantly better than other models. Neural networks learn through multiple layers of abstract features. When faced with class imbalance, they can use weighted training to optimize the detection effect of minority classes. Unlike traditional rule-based or signature-based models, it can automatically learn and adjust weights to improve the ability to predict minority class attacks. Therefore, it is more accurate and stable when dealing with minority class attacks, which is also the inherent reason for its outstanding performance on class-imbalanced datasets.

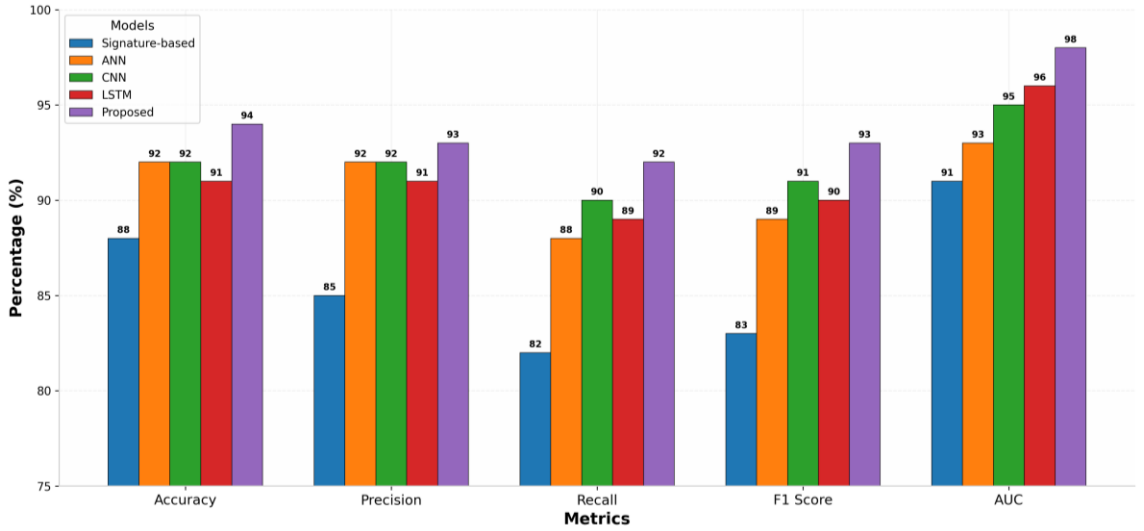

Fig. 3. Performance of different models on class imbalanced datasets

Table 2. Comparison of training time and inference time of different models

Model	Training time (minutes)	Inference time (seconds)	Memory usage (MB)	Model size (MB)	Calculated load (%)
Signature-based model	5	0.2	50	10	25
Artificial Neural Networks (ANN)	12	0.4	120	30	45
Convolutional Neural Network (CNN)	20	0.6	150	60	60
Long Short-Term Memory (LSTM)	25	0.8	180	80	65
The neural network model proposed in this study	15	0.3	100	45	50

Table 2 compares the training time, inference time, memory consumption, model size, and computational load of different models. Although the training time and inference time of the neural network model proposed in this study are longer than those of the traditional signature-based model, the overall performance is still relatively efficient.

In terms of model size and memory consumption, this model has obvious advantages over CNN and LSTM, and is more suitable for deployment in practical scenarios with limited resources. The increase in training time is due to its complex structure, which requires more training data and higher computing resources for learning. However, in the inference stage, the model has a short inference time, which can meet the real-time requirements and achieve a good balance between performance and resource consumption.

The parameters reported in Table 1 exhibit an expected positive correlation with model complexity. However, the relationship is not strictly linear. For instance, while the proposed model (4.8M parameters) is approximately twice the size of the CNN baseline (1.2M parameters), its inference time (0.3 seconds) is only half of that of the CNN (0.6 seconds). This efficiency gain is attributed to the architectural design: the CNN front-end performs aggressive spatial down-sampling, reducing the sequence length before it enters the Transformer encoder. Consequently, the self-attention mechanism operates on a substantially shortened sequence, mitigating the quadratic complexity typically associated with Transformer models. This design choice enables the proposed model to achieve superior detection performance without a proportional increase in inference latency or computational load.

Fig. 4 shows the accuracy of different models for DoS, Probe, R2L and U2R attack types. The neural network model proposed in this study performs well in accuracy for all attack types, and the improvement in accuracy for DoS and Probe attacks is particularly significant. The improvement in accuracy shows that the model has obvious advantages in reducing false positives. The model is trained on a large dataset and learns in depth the subtle differences of each attack mode, thereby reducing the misjudgment of normal traffic. Compared with traditional models, neural networks can better adapt to the complexity and variability of attack modes and avoid false positives. In the face of complex Probe attacks, the neural network model can effectively distinguish between legitimate network detection behaviors and malicious attack activities through deep learning capabilities.

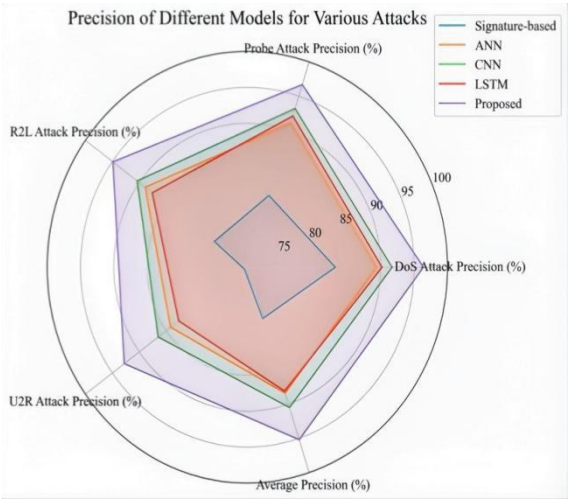

Fig. 4. Comparison of the accuracy of different models under different attack types

Table 3. Comparison of comprehensive evaluation indicators of different models

Model	Accuracy (%)	Recall rate (%)	Accuracy (%)	F1 value (%)	AUC (%)
Signature-based model	89.2	87.6	85.4	86.5	92.5
Artificial Neural Networks (ANN)	93.4	91.2	92.1	91.6	95.3
Convolutional Neural Network (CNN)	94.7	92.5	93.8	93.2	96.1
Long Short-Term Memory (LSTM)	92.1	90.4	91.6	91.0	94.8
The neural network model proposed in this study	97.5	96.3	96.7	96.5	98.2

The superior performance of the proposed model, as demonstrated in Table 3, is attributable to its hybrid architectural design rather than to deep learning in general. While CNN and LSTM baselines also employ multi-layer structures and nonlinear activations, they each possess inherent limitations: CNNs are constrained by fixed receptive fields and struggle to capture long-range dependencies, whereas LSTMs, despite their sequential processing capability, suffer from gradient decay over extended time steps and offer limited parallelism during training. The proposed model overcomes these limitations through two key mechanisms. First, the CNN front-end extracts localized spatial features while reducing input dimensionality. Second, and more critically, the Transformer encoder applies a self-attention mechanism that computes pairwise interactions between all time steps in parallel, enabling the model to capture long-range contextual relationships that are essential for identifying stealthy, multi-step attack patterns. This combination of local feature extraction and global context modeling is the primary driver of the observed performance gains over both CNN and LSTM baselines.

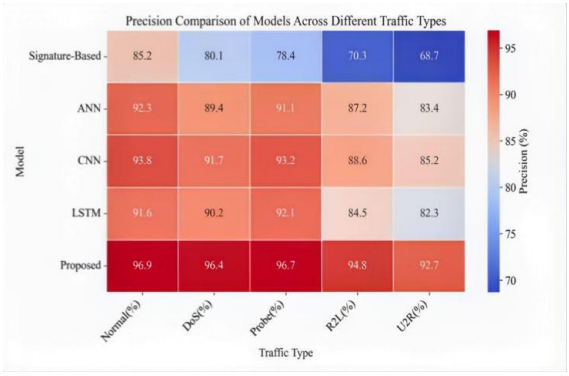

Fig. 5. Performance of different models under different network traffic

Fig. 5 shows the accuracy of different models under normal traffic, DoS traffic, Probe traffic, R2L traffic, and U2R traffic. The neural network model proposed in this study performs significantly better than other models under all types of traffic, especially in the accuracy of normal traffic and DoS traffic. The characteristics of deep neural networks enable them to learn complex traffic patterns from historical data and accurately distinguish normal traffic from attack traffic. In high-dimensional network traffic data, neural networks capture complex spatiotemporal associations through deep hierarchical structures, further improving the accuracy of traffic classification, and can effectively identify malicious attacks and maintain a low false alarm rate under various traffic conditions.

Table 4 compares the rates of false alarms and missed alarms for different models. The model based on neural networks in this study greatly lowers both false alarms and missed alarms, especially false alarms.

This is mainly due to the deep learning ability of the neural network model during the training process. By continuously optimizing weights and activation functions, the model can automatically adapt to the characteristics of different network traffic and reduce false positives. Faced with complex attacks and normal traffic, the neural network model can effectively distinguish between the two and avoid misjudging normal traffic as attack behavior. In addition, through multi-level feature learning, the false negative rate is also significantly controlled, maximizing the accuracy and reliability of detection.

As shown in Fig. 6, different models show obvious performance differences under different network bandwidths. The traditional signature-based model has relatively low accuracy and high false alarm rate under various bandwidths, which reflects its limitations in dealing with network traffic of different bandwidths and its difficulty in adapting to complex and changing network environments. Although ANN, CNN and LSTM have improved in performance, there is still a gap compared with the

neural network model proposed in this study. With its unique architecture and training mechanism, the model in this study can stably maintain high accuracy and low false alarm rate under different network bandwidths, which provides a strong technical guarantee for realizing efficient network intrusion detection in a diverse network environment and has important practical application value and promotion significance.

Table 4. Comparison of false alarm rate and false negative rate of different models

Model	False alarm rate (%)	Missing reporting rate (%)	Correct classification rate (%)	Misclassification rate (%)	Overall error rate (%)
Signature-based model	7.8	12.4	85.5	14.5	13.0
Artificial Neural Networks (ANN)	5.6	8.9	91.4	8.6	7.1
Convolutional Neural Network (CNN)	4.5	7.2	92.3	7.7	6.1
Long Short-Term Memory (LSTM)	6.0	9.6	90.3	9.7	8.2
The neural network model proposed in this study	3.2	5.5	94.8	5.2	4.4

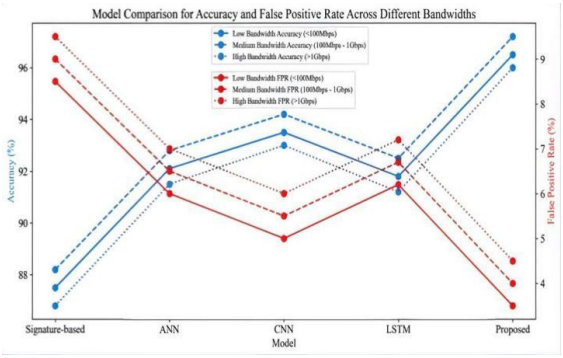

Fig. 6. Performance of different models under different network bandwidths

As shown in Fig. 7, each model's detection performance changes under different attack frequency scenarios. The signature-based model relies on predefined attack features, so it has poor detection effect when facing attacks of different frequencies, with low recall and precision. Models such as ANN, CNN, and LSTM have shown some adaptability, but face limits in improving performance. The neural network model proposed in this study shows excellent detection capabilities under low-frequency, medium-frequency and high-frequency attacks. The high recall rate ensures that the attack behavior can be effectively identified, and the high precision rate ensures the accuracy of the detection results. This advantage enables the model to play a reliable role in actual network security protection, whether facing slowly penetrating low-frequency attacks or high-frequency attacks that break out in a short period of time, to protect network security.

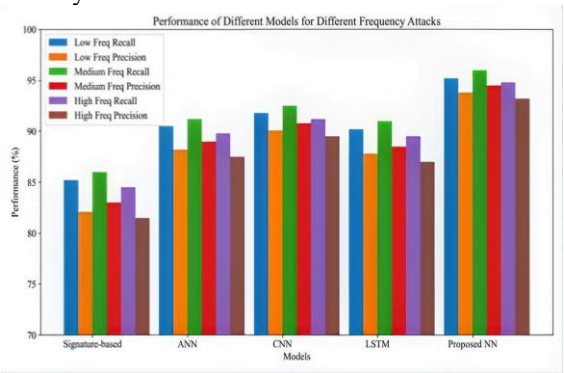

Fig. 7. Detection performance of different models at different attack frequencies

Fig. 8, the training effects of different models on data sets of different sizes show significant differences. The signature-based model is less affected by the size of the data set and has a short training time, but the accuracy improvement is limited, indicating that it has a low degree of dependence on data and lacks the ability to learn complex data patterns. The accuracy of models such as ANN, CNN, and LSTM increases as the dataset becomes larger, but the training time also grows a lot. The neural network model proposed in this study can achieve a high accuracy on a small-scale data set, and under a large-scale data set, not only does it continue to lead in accuracy, but the growth rate of training time is also relatively reasonable. This shows that the model has strong generalization ability and efficient use of data. In practical applications, it can perform well in scenarios with limited data resources or with a large amount of data available for training, providing a better choice for the deployment and application of network intrusion detection systems under different data conditions.

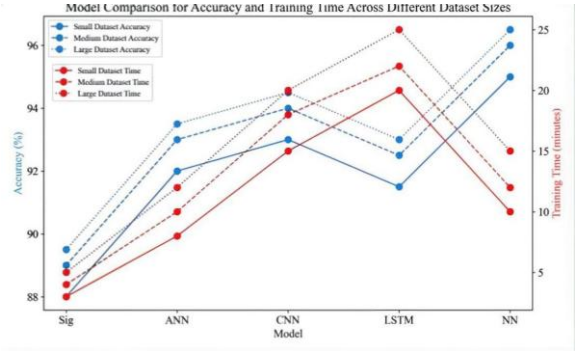

Fig. 8. The training effects of different models under different dataset sizes

4.3. Discussion

From the experimental results, the NIDS built on neural networks in this study performs much better than traditional models and other common neural

network models. It performs well in key indicators such as accuracy, recall, and precision, and shows good adaptability and stability under different network bandwidths, attack frequencies, and data set sizes.

The reason why this model can achieve such results is mainly due to its innovative architecture design. Combining the Transformer encoder with the traditional neural network, the Transformer's self-attention mechanism can well capture long-term dependencies in network traffic, giving the model a deeper understanding of complex attack patterns.

At the same time, the feature extraction module uses CNN for preliminary processing, effectively compressing the data dimension and extracting key features, providing a high-quality data foundation for subsequent processing, enhancing the model's ability to capture various attack features, and can accurately identify various attack types such as DoS and Probe.

In terms of processing class imbalanced data sets, the model automatically adjusts weights to improve the detection capability of minority attacks through multi-layer abstract feature learning and weighted training, effectively solving the difficulties of traditional models in this regard. However, the model is not perfect. Although it can meet the real-time requirements in the inference stage, the training time is still increased compared to the traditional signature-based model, which is due to its complex structure and large amount of training data requirements. Future research can consider optimizing the training algorithm to further shorten the training time. At the same time, despite the excellent performance of the model, the "black box" characteristics of the neural network still exist. How to improve the interpretability of the model and help security experts better understand how the model makes decisions is also a key direction for future work.

5. CONCLUSION

This study aims to address the increasingly complex network attack problem in network security and proposes a NIDS built on neural networks. The Transformer encoder is integrated into the traditional neural network architecture, and a CNN is applied to extract features, building a model that can effectively process complex network traffic data. The model achieves accurate identification of network intrusion behavior through the collaborative work of various modules. The experimental results indicate that the model in this study performs well. In terms of key indicators, including accuracy, recall, and precision, it greatly outperforms traditional models and other common neural network models. When facing different network bandwidths, attack frequencies, and data set sizes, the model shows good adaptability and stability. Especially when dealing with class imbalanced data sets, the model can effectively improve the detection ability of minority attacks, provide more reliable protection for network security protection, and strongly promote the development of network intrusion detection technology. However, this study also has certain limitations. The model training time is relatively long, which is mainly due to its complex structure and large amount of training data requirements, which limits the rapid deployment and update of the model in practical applications. At the same time, the "black box" characteristics of neural networks lead to a lack of transparency in the model decision process, which affects the trust of security experts in model decisions. Future studies may focus on improving training algorithms, for example by using more efficient ways to allocate computing resources and shorten training time. For model interpretability issues, visualization technology can be combined or a special interpretive model can be developed to make the model decision process more transparent, thereby enhancing the application value of the model in actual network security protection.

Source of funding: *This study did not receive external funding.*

Declaration of competing interest: *The author declares no conflict of interest.*

REFERENCES

1. Tian WX, Shen YP, Guo N, Yuan J, Yang YQ. VAE-WACGAN: An improved data augmentation method based on VAEGAN for intrusion detection. Sensors. 2024; 24(18). https://doi.org/10.3390/s24186035.
2. Yu J, Ye XJ, Li HB. A high precision intrusion detection system for network security communication based on multi-scale convolutional neural network. Future Generation Computer Systems-the International Journal of Escience .2022; 129:399-406. https://doi.org/10.1016/j.future.2021.10.018.
3. Luo J, Zhang YY, Wu YN, Xu Y, Guo XY, Shang BX. A multi-channel contrastive learning network based intrusion detection method. Electronics. 2023;12(4). https://doi.org/10.3390/electronics12040949.
4. Chellammal P, Malarchelvi SK, Reka K, Raja G. Fast and effective intrusion detection using multi-layered deep learning networks. International Journal of Web Services Research. 2022;19(1). https://doi.org/10.4018/ijwsr.310057.
5. Fang WJ, Tan XL, Wilbur D. Research on machine learning method and its application technology in intrusion information security detection. Journal of Intelligent & Fuzzy Systems. 2020; 38(2):1549-58. https://doi.org/10.3233/jifs-179518.
6. Hu XY, Gao WJ, Cheng G, Li RD, Zhou YY, Wu H. Toward early and accurate network intrusion detection using graph embedding. IEEE Transactions on Information Forensics and Security. 2023; 18:5817-31. https://doi.org/10.1109/tifs.2023.3318960.
7. Fan ZJ, Cao ZW. Method of network intrusion discovery based on convolutional long-short term memory network and implementation in VSS. IEEE Access. 2021; 9:122744-53. https://doi.org/10.1109/access.2021.3104718.

8. Siddiqi MA, Pak W. Tier-based optimization for synthesized network intrusion detection system. IEEE Access. 2022; 10:108530-44. https://doi.org/10.1109/access.2022.3213937.
9. Zou L, Luo XM, Zhang Y, Yang X, Wang XW. HC-DTTSVM: A network intrusion detection method based on decision tree twin support vector machine and hierarchical clustering. IEEE Access. 2023; 11: 21404-16. https://doi.org/10.1109/access.2023.3251354.
10. Sun Y, Que HK, Cai QQ, Zhao JM, Li JR, Kong ZM, et al. Borderline SMOTE algorithm and feature selection-based network anomalies detection strategy. Energies. 2022; 15(13). https://doi.org/10.3390/en15134751 .
11. Zuo XJ, Chen Z, Dong LM, Chang J, Hou BT. Power information network intrusion detection based on data mining algorithm. Journal of Supercomputing. 2020; 76(7):5521-39. https://doi.org/10.1007/s11227-019-02899 -2.
12. Wan XH, Xue G, Zhong YM, Wang ZC. Separating prediction and explanation: an approach based on explainable artificial intelligence for analyzing network intrusion. Journal of Network and Systems Management. 2025; 33(1). https://doi.org/10.1007/s10922-024 -09891-z.
13. Wang H, Cao ZJ, Hong B. A network intrusion detection system based on convolutional neural network. Journal of Intelligent & Fuzzy Systems. 2020; 38(6):7623-37. https://doi.org/10.3233/jifs-179833.
14. Wang YW, Xu L, Liu WL, Li RZ, Gu JJ. Network intrusion detection based on explainable artificial intelligence. Wireless Personal Communications. 2023;131(2):1115-30. https://doi.org/10.1007/s11277-023-10472-7.
15. Ayyagari MR, Kesswani N, Kumar M, Kumar K. Intrusion detection techniques in network environment: a systematic review. Wireless Networks. 2021;27(2):1269-85. https://doi.org/10.1007/s11276-020-02529-3.
16. Spiekermann D, Eggendorfer T, Keller J. Deep learning for network intrusion detection in virtual networks. Electronics. 2024;13(18). https://doi.org/10.3390/electronics13183617.
17. Geng ZQ, Li XM, Ma B, Han YM. Improved convolution neural network integrating attention based deep sparse auto encoder for network intrusion detection. Applied Intelligence 2025; 55(2). https://doi.org/10.1007/s10489-024-05872-6.
18. Qazi EU, Faheem MH, Zia T. HDLNIDS: Hybrid deep-learning-based network intrusion detection system. Applied Sciences-Basel. 2023;13(8). https://doi.org/10.3390/app13084921.
19. Wang YL, Li J, Zhao W, Han ZJ, Zhao H, Wang L, et al. N-STGAT: Spatio-temporal graph neural network based network intrusion detection for near-earth remote sensing. Remote Sensing. 2023;15(14). https://doi.org/10.3390/rs15143611.
20. Babu DS, Ramakrishnan M. Enhanced lion optimization algorithm and deep belief network for intrusion detection with SDN enabled IoT networks. Journal of Intelligent & Fuzzy Systems. 2023;45(4): 6605-15. https://doi.org/10.3233/jifs-232532.
21. Wen LB. Cloud computing intrusion detection technology based on BP-NN. Wireless Personal Communications. 2022;126(3):1917-34. https://doi.org/10.1007/s11277-021-08569-y.

Baoxing XIE received his B.S. degree in Science & Technology Information from Zhengzhou University of Aeronautics in 2001. His research interests include computer networks, network security, and big data processing.
e-mail:
baoxing_xie@outlook.com