Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 15, No. 3, 2026

Improved CNN-Transformer and SVM Framework for Intrusion Detection

Minyi Jin *

International College, Beijing University of Posts and Telecommunications, Beijing, China
*** Corresponding author Email:** 2022213362@bupt.cn

Abstract: Facing the large-scale deployment of Internet of Things in smart homes, smart cities, and industrial scenes, the characteristics of heterogeneous terminals, limited resources, and diverse protocols make it more vulnerable to network attacks such as malicious control, DDoS, and data theft, and an intrusion detection system has become an important aspect of IoT security protection. To cope with the evolution of attack morphology and the complexity of traffic characteristics, this paper proposes an intrusion detection framework, STIF, combining an improved CNN-Transformer model and a support vector machine (SVM). The framework aims to address the problems of insufficient modeling of local details and limited characterization of long-range dependence in traditional convolutional neural networks (CNNs) and Transformers for IoT intrusion detection. In the pre-training section, depthwise separable convolution is introduced to reduce parameter count and computational overhead, and the Transformer decoder's structure is lightweight and optimized to meet the edge deployment requirements better. In the decision-making section, SVM is used to determine a more robust decision boundary, thereby improving the recognition of minority attacks. The framework is verified on the X-IIoTID dataset, and the experimental results show that the overall detection accuracy reaches 0.9860 and that it effectively identifies small-scale categories.

Keywords: Internet of Things; Intrusion Detection; Convolutional Neural Network; Transformer; Support Vector Machine.

1. Research Aim and Significance

The Internet of Things (IoT) connects sensors, actuators, and embedded terminals to the cloud-edge-end network to enable perception, control, and data services for the physical world. With the rapid popularization of smart homes, smart cities, the Internet of Vehicles, wearable devices, and medical devices, IoT systems are characterized by a large number of terminals, heterogeneous protocols, distributed deployment, and limited equipment resources, which significantly expand the attack surface. Common threats include botnets and DDoS, malicious firmware and remote control, data theft and privacy disclosure, protocol spoofing and replay, etc. To improve the usability and data security of IoT systems, it has become a key technology in the Internet of Things security protection system to build an efficient Intrusion Detection System (IDS) that continuously monitors network traffic and terminal behavior and identifies anomalies.

The aim and significance of this paper are as follows:

(1) IoT system security safeguarding

By developing the STIF framework, a more efficient and accurate intrusion detection solution is provided for the Internet of Things environment to address increasingly serious security threats. By improving the security capabilities of IoT, the risks of data leakage and cyberattacks can be effectively reduced, and the continuity and service reliability of key businesses can be guaranteed, thereby promoting the safe landing and large-scale application of Internet of Things technology.

(2) Reducing manual intervention and improving efficiency

Traditional intrusion detection technology relies on manual operation, which is not only costly but also susceptible to human resource constraints and operational errors. The framework automates the analysis of network traffic data, reduces reliance on professional labor, and improves detection efficiency and accuracy.

(3) Improving the existing model for the IoT environment

Given the actual Internet of Things environment, in the traditional CNN model, using depthwise separable convolutions instead of a standard convolutional layer can significantly reduce the number of parameters and computational requirements, offering advantages for some IoT devices with limited computing resources. The decoder structure of the traditional Transformer model is improved, and the self-attention mechanism is eliminated in the decoding stage, thereby avoiding the extraction of information from the tag. At the same time, combined with the improved CNN-Transformer, it leverages the advantages of CNNs in extracting local features to compensate for the Transformer's shortcomings in this regard.

1.1. Research Status at Home and Abroad

Many researchers have proposed various machine learning and deep learning techniques to enhance the intrusion detection system's ability to identify hostile behavior. Xu Dongfang and his team proposed an intrusion detection framework combining CNNs, bidirectional long- and short-term memory networks, and extreme gradient boosting, aiming to address the limitations of insufficient feature extraction and inaccurate multi-class classification in intrusion detection methods [1]. Lilhore et al. effectively improved the intrusion detection model's prediction accuracy by employing the Grey Wolf Optimizer to fine-tune key CNN parameters, including pool size, kernel size, number of filters, epochs, and batch size [2]. The research of Li Di et al. showed that by combining CNN with long-term and short-term memory networks, it is possible not only to effectively identify and deal with the spatial characteristics, but also deal with the time dependence in time series data, so as to understand and predict the dynamic changes of network traffic more comprehensively. At the same time, the attention

mechanism is introduced to further enhance the model's ability to focus on key information, enabling the system to identify potential threats more quickly in a complex network environment, thereby achieving higher detection accuracy [3].

In addition, other studies, such as Yin Y et al., combined MLP and random forest technology in their work. Through this hybrid method, they significantly enhance the model's ability to classify different types of attacks, enabling it to deal with various network anomalies more effectively [4]. Similarly, Süzen used a hybrid multi-level DBN algorithm to build an intrusion detection system, which effectively detects and classifies abnormal data by combining the DBN's last layer with a Softmax classifier. The effectiveness of this method has been verified using a dataset from a real industrial control system [5]. These studies not only advance the development of intrusion detection technology but also provide additional protection strategies and tools for the security of the Internet of Things.

Research shows that most existing intrusion detection systems fully capture data features by integrating multiple functions rather than relying on a single technology. In addition, the effect of the intrusion detection model depends largely on its feature extraction ability [9]. In view of this, given the characteristics of Internet of Things devices and network traffic, this paper proposes an intrusion detection framework, STIF, combining an improved CNN-Transformer model and SVM, aiming to improve the accuracy and efficiency of intrusion detection in the IoT environment. Through this research, we can not only deepen our understanding of the security threats posed by the Internet of Things but also provide a novel and effective technical means for network security protection, with important theoretical significance and application value.

1.2. Main Research Contents

1.2.1. Main Research Contents

(1) Depthwise separable convolution is introduced into the traditional CNN model to replace the standard convolution layer. This improved convolution method can greatly reduce the model's parameter count and computational requirements, enabling it to achieve significant performance gains on IoT devices with limited computing resources.

(2) In order to deal with abnormal network traffic more effectively, the decoder structure of the traditional Transformer model is improved, so that the model can identify and classify abnormal network traffic more accurately.

(3) Combining the improved CNN and Transformer model, we make use of the advantages of CNN in extracting local features to make up for the shortcomings of the Transformer in this respect. This fusion strategy significantly improves the effectiveness and accuracy of the intrusion detection model when applied to Internet of Things data.

(4) The simulation experiment on the intrusion dataset X-IIoTID designed for the IoT environment verifies the superiority of the STIF framework. The experimental results show the advantages of STIF in identifying a small number of sample categories and achieving high accuracy.

2. Related Technologies

2.1. Convolutional Neural Network and Lightweight Convolution

CNNs are good at extracting local pattern features from time-series traffic. To adapt to IoT equipment with limited resources, this paper adopts depthwise separable convolution to reduce parameter count and computational cost, improving reasoning efficiency while maintaining the ability to express features [7].

2.2. Transformer and Self-attention Mechanism

Transformer can capture long-range behavior patterns and complex attack link characteristics by directly modeling the dependencies between arbitrary positions in the sequence through self-attention [8]. Combining a lightweight structure with engineering cutting can further reduce deployment expenses.

2.3. Support Vector Machine (SVM)

SVM constructs a classification hyperplane based on the principle of maximum margin, which has good generalization performance in high-dimensional feature spaces and small- to medium-sized sample scenarios. In this paper, SVM is introduced into the decision-making layer for robust discrimination of depth features [9].

2.4. CNN-Transformer Fusion

In the STIF framework, the CNN extracts low-level features such as local statistics and sudden changes, while the Transformer models the global dependencies among local features. Complementing each other helps to cover both short-term anomalies and long-term behavior patterns, thus improving the accuracy and robustness of intrusion detection [10].

3. Framework Design

The STIF intrusion detection framework presented in this paper is mainly divided into a pre-training section and a decision-making section. In the pre-training section, the CNN model processes the one-dimensional traffic data using a depthwise separable and extracts key local features. The transformer model captures the global dependencies in the input data through a self-attention mechanism. Meanwhile, the self-attention mechanism in the decoding section is removed, and this adjustment is better suited for processing and identifying abnormal traffic data in an IoT environment. Finally, it is the decision-making section of STIF. After preliminary feature extraction and in-depth global analysis, all data features are summarized and sent to the SVM. Here, SVM makes the final judgment based on the integrated characteristics.

3.1. Pre-training Section Design

3.1.1. CNN Section

In the pre-training stage of STIF, CNN is mainly used to extract local features from the input network traffic data. This process involves several key steps, including a depthwise separable convolution operation, an activation function, and pooling. A convolution kernel with a specific size and zero-padding is used to maintain the feature dimension, thereby minimizing edge information loss.

(1) Zero-padding strategy

Convolution usually reduces the dimensionality of data, especially in time-series data processing, which may lead to the loss of edge information. To prevent this situation and preserve the original time series length, the STIF framework

uses a zero-padding strategy before performing the convolution operation.

Assuming that the dimension of network traffic event X is $T \times N$, where T is the number of time steps, and N is the number of features in X. If zero-padding is performed at the beginning and end of the sequence, the processed data dimension becomes (T, N+2).

(2) Depthwise separable convolution operation

In the context of network traffic data, each channel can be regarded as an independent characteristic dimension, such as packet size, time interval, and so on. Deep convolution can effectively capture the spatial features of each feature without interference from other features by processing each channel independently. Deep convolution performs convolution independently for each expanded channel. It uses a convolution kernel $k[n]$ to slide over the expanded time series, performing pointwise multiplication and summation to extract features.

In the depth convolution section, assuming that $x[t,n]$ represents the flow data point of time t and characteristic n, and the convolution kernel is $k[n]$, then the output $y[t,n]$ of the depth convolution is shown in Formula (1).

$$y[t,n] = \sum_{m=-M}^{M} x[t+m,n] \cdot k[m,n] \quad (1)$$

Where M is half the width of the convolution kernel $k[n]$, and the index of t is from M to $T+M-1$, which ensures that the convolution kernel completely covers the input data at any time point t.

In the point-by-point convolution section, the output of the deep convolution is integrated in the channel dimension, and a 1×1 convolution kernel is used. This can be regarded as a linear transformation that merges information from different characteristic channels. According to the above, y[t,n] is the output of the deep convolution, and the result $z[t]$ after point-by-point convolution can be expressed as (2).

$$z[t] = \sum_{n=1}^{N} y[t,n] \cdot w[n] \quad (2)$$

Where $w[n]$ is the weight of a 1×1 convolution kernel.

(3) Activation function

Although convolutional layers can extract useful features, the expressive power of the entire network is limited by their linear operations. The purpose of the activation function is to introduce nonlinearity, enabling the design framework to learn more complex patterns. ReLU is the activation function used in CNN in this paper, and its definition is shown in Formula (3).

$$R(z) = max(0, z[t]) \quad (3)$$

(4) Pool operation

The role of the pooling layer in a neural network is to reduce the size of the feature map while retaining key features. By implementing the max pooling strategy, this layer subdivides the input feature map into several independent rectangular blocks. It selects the maximum value from each block for output, effectively reducing the data volume and highlighting the main features. The design is shown in Formula (4).

$$P[t] = max(a[t - M_P : t + M_P]) \quad (4)$$

Through the above steps, the CNN section of the STIF framework not only optimizes feature extraction but also greatly enhances the nonlinear expression ability of the network through the use of point-by-point convolution and activation function, and reduces the feature dimension through pooling, providing high-quality input for the subsequent Transformer model and SVM decision-making layer processing. These operations together ensure that the system can effectively and accurately process and analyze the complex data needed for intrusion detection.

3.1.2. Transformer Section

In the pre-training stage of the STIF framework, the Transformer architecture plays a vital role, particularly in its encoder. It can capture rich global features from the input data through a complex self-attention mechanism and network layer stacking.

The transformer section receives the feature information $P[t]$ extracted from the CNN, and first performs a position coding operation on it.

(1) Multi-head self-attention mechanism

This key component enables the Transformer model to detect interdependencies across different sections of the input sequence. Although each "head" deals with the same input data, they can learn and extract information in multiple subspaces by applying different weight configurations. The output of the self-attention mechanism is shown in Formula (5).

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (5)$$

Among them, Q, K, and V represent query, key, and value, respectively. d_k is the dimension of the key and helps to stabilize the gradient. Multi-head attention is a linear transformation after splicing the outputs of different heads.

(2) Multi-head attention

The Transformer model enhances its processing ability by using multiple parallel attention heads, each operating independently in its own representation subspace, enabling it to capture various features of the input data and improve the overall expressive power of the model.

$$MultiHead(Q,K,V) = Concat(head1,...,headh)Wo \quad (6)$$

In $head_i = Attention(Q W_i^Q, K W_i^K, V W_i^V)$, each head corresponds to a set of independent weight matrices W_i^Q, W_i^K, and W_i^V, and finally, the outputs of different heads are spliced by another set of weights W^o to obtain the attention values of all "heads".

(3) Feedforward network

Each attention layer is followed by a feedforward network that applies the output of each position independently and equally, providing additional nonlinear transformation power. As shown in formula (7).

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \quad (7)$$

Among them W_1, b_1, W_2 and b_2 are network parameters, and the ReLU activation function is used, which increases the model's complexity.

3.2. Decision-Making Section Design

In the STIF framework, the pre-trained and initialized classification data are fed into the SVM for further training.

After training, the SVM generates a score vector for each category using the decision function G(Z), where Z is the feature vector output by the pre-training module. The expression of this decision function is shown in Formula (8).

$$G(Z) = sign(v_m Z + c_m) \quad (8)$$

Where v_m and c_m are a weight vector and an offset term, respectively.

In addition, the model uses *the argmax function to determine the final prediction's category label,* as shown in Formula (9).

$$Pre\_lable = argmax(G(Z)) \quad (9)$$

The framework combines the classification capabilities of Softmax and SVM to further improve the model's accuracy. This method not only effectively distinguishes different kinds

of samples but also enhances the model's generalization to unseen data by increasing the distance between the hyperplane and the nearest data point.

4. Experiments

4.1. Experiment and Analysis

In this paper, X-IIoTID, an intrusion dataset tailored for the IoT environment, is used to evaluate the performance of the STIF framework and compare it with CNN, RNN, CNN-RNN, and the improved CNN-Transformer model. Through comparative analysis, the purpose is to demonstrate the advantages of the STIF framework across various performance indicators and to verify the effectiveness of the designed framework.

4.2. Preprocessing of the Dataset

The X-IIoTID dataset used in the experiment contains multi-view features, such as network traffic, host resources, and system logs, which are suitable for evaluating the intrusion detection capability in complex IoT scenarios. This paper selects a subset of 40,000 instances and 62 features per instance. In the preprocessing stage, the invariant feature columns such as "is_SYN_with_RST" are removed, and "class2" is selected as the experimental label to capture fine-grained attacks. To enhance model training stability and adapt to a one-dimensional convolutional network, all features are standardized, and the samples are finally sorted into tensors with the shape (40,000, 1, 57), ensuring that the convolutional layer can effectively extract the data.

4.3. Parameter Settings of Frame and Contrast Model

4.3.1. Parameter Setting STIF of the Framework

In the pre-training section, the CNN contains 7 convolutional blocks. A depth-separable convolution with a kernel size of 2 is used to analyze slight changes in time series. Padding is set to 1 to preserve edge information, and the number of channels is multiplied by the number of blocks layer by layer. The pooling layer uses max pooling with a step size of 2 to ensure the output shape adapts to subsequent processing. The position encoder in the Transformer section is set to 200, and a Dropout of 0.1 is used to prevent overfitting. The encoder has 2 layers, and features are extracted in parallel via a double-headed attention mechanism. Dropout is set to 0.2 to improve generalization. The improved decoder converts high-dimensional features into 128 dimensions, then passes them through a five-layer fully connected layer, introduces nonlinearity via the ReLU function, and finally outputs the probabilities of 10 attack types via a Softmax layer.

4.3.2. Parameter Setting of the Comparison Model

The parameters of the improved CNN-Transformer model are consistent with the pre-training section of STIF, aiming to verifying the effectiveness of data processing before classification. In the CNN-RNN model, the CNN section maintains the same configuration, and the RNN section has 150 hidden units across two levels; its output is then classified by three fully connected layers after compression. The pure CNN model directly connects three fully connected layers behind the convolution layer for dimension compression and decision-making. The pure RNN model reshapes the dataset to (40,000, 57, 1), and its component parameters are fully consistent with the corresponding sections of the CNN-RNN model to ensure comparability across experiments.

4.4. Model Evaluation Indicators

A confusion matrix can be used to visualize the performance of machine learning algorithms, especially in classification problems [12]. It shows the model's performance across various categories by comparing its predicted results with the actual results. The confusion matrix is very useful for understanding the model's concrete performance, because it not only shows correct predictions but also the number of classification errors. For the binary classification problem, the confusion matrix is shown in Table 1.

Table 1. Confusion Matrix

	Predicted as positive	Predicted as negative
Actually, it is positive.	True positive (TP)	False negative (FN)
Actually, it is negative.	False positive (FP)	True negative (TN)

A confusion matrix can be used to calculate a variety of performance indicators; the indicators used in this paper are as follows.

(1) Accuracy, which represents the proportion of correct predictions (including positive and negative categories) among all predictions. The calculation method is shown in Formula 10.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{10}$$

(2) Precision, which indicates the proportion of positive samples predicted by the model, reflects the accuracy of positive samples predicted by the model. The calculation method is shown in Formula 11.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{11}$$

(3) Recall, which represents the proportion of positive samples in the prediction, reflects the accuracy of the model in predicting positive samples. The calculation method is shown in Formula 12.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{12}$$

(4) F1 score, which is the harmonic mean of accuracy and recall, is mainly used to measure the accuracy of the classification model, and the calculation method is shown in Formula 13.

$$\text{F1} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision+Recall}} \tag{13}$$

In the STIF framework, a confusion matrix is used to evaluate the classification accuracy and error types of each classification model. By comparing the confusion matrices and related indicators across models, we can better understand which models perform better at recognizing specific categories and which categories may be frequently misclassified due to model deviation.

4.5. Analysis and Discussion of Experimental Results

4.5.1. Performance Comparison of STIF Kernel Functions

In order to explore the best performance of the STIF framework, four kernel functions, Linear, RBF, Sigmoid, and Poly, are tested in the decision-making section of SVM, and the results are shown in Table 2. In this table, 10 attack categories (from Normal to crypto-ransomware) are

represented by numbers 0-9, and the accuracy, precision, recall rate, and F1 score are abbreviated as Acc, PRE, RC, and F1, respectively. The experimental results show that the RBF kernel performs best, with excellent nonlinear processing ability, and the overall accuracy reaches 0.9860. Especially in the "C&C" and "crypto-ransomware" categories, the RBF kernel's detection rate reaches 1.0, effectively handling complex attack data structures.

Table 2. Performance Comparison of STIF Using Different Kernel Functions

	Linear			RBF			Sigmoid			Poly		
	PRE	RC	F1	PRE	RC	F1	PRE	RC	F1	PRE	RC	F1
0	0.932	0.960	0.946	0.937	0.965	0.951	0.774	0.782	0.778	0.896	0.956	0.925
1	1.000	0.995	0.998	1.000	0.995	0.998	0.977	0.977	0.977	0.983	0.987	0.985
2	1.000	1.000	1.000	0.999	0.999	0.999	0.996	0.985	0.990	1.000	0.998	0.999
3	0.959	0.886	0.921	0.989	0.935	0.961	0.964	0.810	0.880	0.943	0.892	0.917
4	0.989	0.988	0.989	0.988	0.990	0.989	0.873	0.915	0.893	0.982	0.993	0.988
5	0.998	1.000	0.999	1.000	1.000	1.000	0.990	1.000	0.995	0.998	0.992	0.995
6	0.953	0.948	0.951	0.982	0.957	0.969	0.912	0.885	0.899	0.973	0.924	0.948
7	0.990	0.987	0.989	0.982	1.000	0.991	0.948	0.965	0.956	0.987	0.984	0.985
8	0.998	0.997	0.997	0.996	0.998	0.997	0.988	0.975	0.982	0.996	0.996	0.996
9	1.000	1.000	1.000	1.000	0.333	0.500	0.000	0.000	0.000	1.000	0.333	0.500
Acc	0.9815			0.9860			0.9315			0.9773		

Compared with the RBF kernel, the linear kernel is similar in some categories but inferior in most. For example, in the "Exploitation" detection, the F1 score for the RBF kernel (0.961) is significantly higher than that for the Linear kernel (0.921), indicating its advantage in nonlinear mapping.

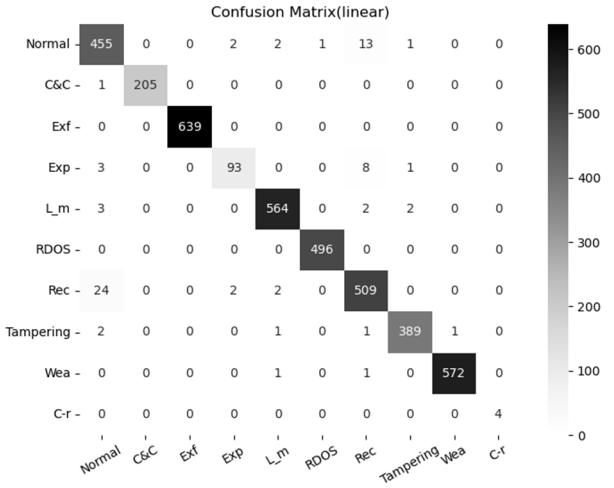

Figure 1. Confusion Matrix for STIF with the Linear Kernel Function

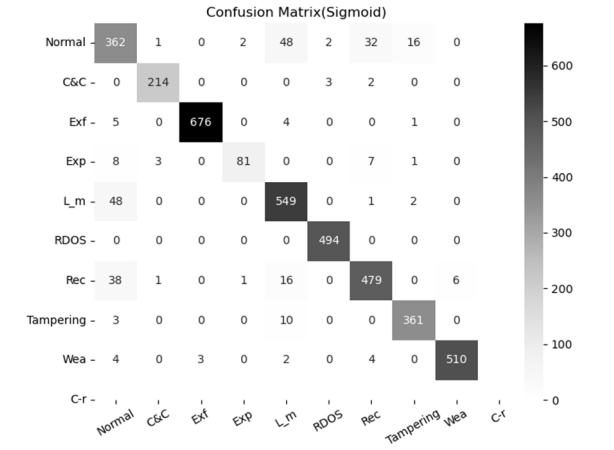

Figure 2. Confusion Matrix for STIF with the Sigmoid Kernel Function

The sigmoid kernel has the worst experimental performance due to its limited ability to handle complex nonlinear problems, with an accuracy of only 0.9315, and all indicators in the "crypto-ransomware" category are 0, so it cannot effectively identify this kind of attack. The performance of Poly core is between RBF and Sigmoid. Although it is excellent in some categories, such as "Exfiltration" (F1 score of 0.999), its overall stability and comprehensive performance are still inferior to those of the RBF core. The confusion matrices for the different kernel functions used by the STIF framework are shown in Figures 1 to 4.

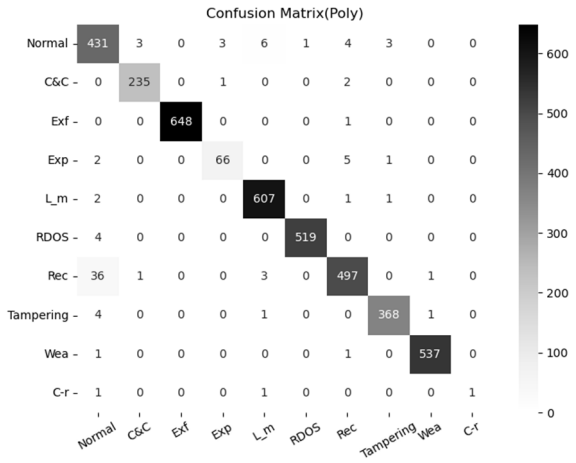

Figure 3. Confusion Matrix for STIF with the Poly Kernel Function

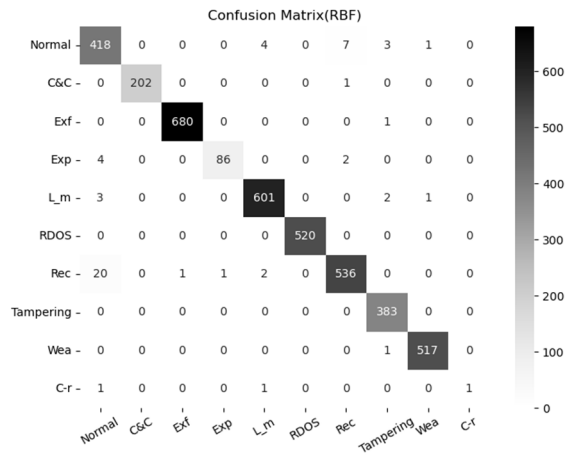

Figure 4. Confusion Matrix for STIF with the RBF Kernel Function

4.5.2. Performance Comparison of the Control Group Model

The performance comparison of the control model is shown in Table 3.

Table 3. Performance Comparison of Control Group

	(Improved) CNN-Transformer			CNN-RNN			CNN			RNN		
	PRE	RC	F1	PRE	RC	F1	PRE	RC	F1	PRE	RC	F1
0	0.961	0.980	0.971	0.886	0.926	0.906	0.874	0.909	0.891	0.915	0.966	0.940
1	1.000	1.000	1.000	0.991	0.996	0.993	0.991	1.000	0.996	0.996	1.000	0.998
2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
3	0.986	0.911	0.947	0.952	0.778	0.856	0.921	0.774	0.841	0.977	0.934	0.955
4	0.996	1.000	0.998	0.992	0.995	0.993	0.981	0.995	0.988	0.978	0.998	0.988
5	1.000	0.998	0.999	0.993	0.996	0.995	0.994	0.992	0.993	0.998	1.000	0.998
6	0.973	0.965	0.969	0.922	0.905	0.913	0.937	0.923	0.930	0.986	0.918	0.951
7	1.000	1.000	1.000	0.995	0.995	0.995	0.997	0.995	0.996	0.989	0.997	0.993
8	0.998	1.000	0.999	0.991	0.998	0.995	0.996	0.996	0.996	0.994	0.998	0.996
9	1.000	1.000	1.000	0.000	0.000	0.000	0.500	1.000	0.667	1.000	1.000	1.000
Acc	0.9908			0.9697			0.9710			0.9812		

The experimental results show that the improved CNN-Transformer model performs best in the control group, with an accuracy of 0.9908. Analysis of the data in Table 3 shows that the PRE, RC, and F1 scores of the model in most classifications (such as 1, 2, 7, and 9) all reach 1.0, indicating its advantages in dealing with small samples and complex-structured data. Under the same CNN parameters, the accuracy of the CNN-Transformer is higher than that of the CNN-RNN, demonstrating that the Transformer is superior to the RNN in capturing global features. In contrast, CNN-RNN performs poorly on classification 9, reflecting its limitations in capturing long-term dependencies. However, the overall accuracy of the independent CNN and RNN models (0.9710 and 0.9812, respectively) is lower than that of the improved model. Experiments further demonstrate the superiority of the improved model in global feature capture and small-sample recognition, with high recall rates and stability.

4.5.3. Comparison between Benchmark Model and STIF

Table 4 compares the detection rates of STIF and benchmark models on X-IIoTID datasets. Experiments show that the detection rate of STIF for attack types such as “C&C”, “Weaponization” and “crypto-ransomware” all reaches 1.0, and the consistency is significantly better than that of the comparison model. The detection performance of STIF is more balanced across attack categories, effectively avoiding performance fluctuations and enhancing the reliability of practical applications. Especially in identifying small-sample attacks such as “crypto-ransomware”, STIF’s detection rate ranks first, highlighting its ability to capture rare samples and underscoring its significance for defending against harmful minority attacks.

Table 4. Comparison Between Benchmark Model and STIF

Model	1	2	3	4	5	6	7	8	9
DT	0.887	0.867	0.877	0.924	0.930	0.983	0.994	0.995	0.978
NB	1.000	0.942	0.826	0.093	0.980	0.055	0.891	0.988	0.886
KNN	0.812	0.721	0.965	0.912	0.918	0.968	0.786	0.992	0.976
SVM	0.845	0.728	0.589	0.986	0.945	0.932	0.921	0.996	0.989
LR	0.512	0.412	0.683	0.951	0.976	0.831	0.861	0.989	0.988
DNN	0.768	0.998	0.833	0.864	0.998	0.998	0.971	0.997	0.934
GRN	0.892	0.998	0.761	0.972	0.922	0.956	0.998	0.979	0.932
STIF(RBF)	1.000	0.987	0.936	0.924	0.982	0.934	0.989	1.000	1.000

Finally, STIF not only performs well in a single category but also maintains a stable, efficient overall detection rate, especially when dealing with complex or rare attack types. By accurately and stably identifying various network attacks, STIF provides a strong technical support for the field of network security. The above experimental results not only verify the effectiveness of STIF design but also show its reliability and wide applicability in practical deployment.

5. Conclusion

Targeting IoT security scenarios, this paper proposes STIF, an intrusion detection framework that integrates an enhanced CNN-Transformer feature extractor with an SVM discriminator. The framework employs depthwise separable convolution to reduce computational overhead and utilizes an attention mechanism to model long-range dependencies. Evaluated on the X-IIoTID dataset, STIF achieves high detection performance, attaining an overall accuracy of 0.9860 with the RBF kernel. Comparative experimental results demonstrate that the framework exhibits strong robustness against multi-category attacks and maintains a relative advantage in detecting minority classes.

References

[1] Xu Dongfang, Li Qi, Peng Kaibo. Intrusion detection system based on CNN-BLSTM-XGB [J]. *Computer Engineering and Design*, 2024, 45(3): 676-683.

[2] Yin Y, Cheng L, Geng L, et al. An effective feature selection method for network intrusion detection based on improved genetic algorithm and recursive feature elimination[J]. *Journal of Big Data*, 2023, 10(1): 15. DOI:10.1186/s40537-023-00689-w.

[3] Li Jiafeng, Xue Xiao, Wan Jinbin, et al. research on network intrusion detection method of industrial control system based on CNN-LSTM-Attention [J]. *Thermal Power Generation*, 2024,53 (5): 61-70. doi: 10.19666/J. rlfd.20016.00000000106

[4] Süzen A A. A multi-level hybrid-DBN based intrusion detection system for IoT networks[J]. *IET Communications*, 2021, 15(12): 1575-1587.

[5] Yang Xiaowen, Cui Zewen, Ma Yulin. Attention-based intrusion detection method using CNN-BiGRU[J]. *Information Security Research*, 2024, 10(3): 202-208.

[6] Isong B, Kgote O, Abu-Mahfouz A. Insights into Modern Intrusion Detection Strategies for Internet of Things Ecosystems[J]. *Electronics*, 2024, 13(12): 2370. DOI: 10. 3390/ electronics13122370.

[7] Howard A G, Zhu M, Chen B, et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications [EB/OL]. arXiv:1704.04861, 2017.

[8] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017: 6000-6010.

[9] Cervantes J, García-Lamont F, Rodríguez-Mazahua L, et al. A comprehensive survey on support vector machine classification: Applications, challenges and trends[J]. *Neurocomputing*, 2020, 408: 189-215. DOI:10.1016/j.neucom.2019.10.118.

[10] Long Z, Yan H, Shen G, et al. A Transformer-based network intrusion detection approach for cloud security[J]. *Journal of Cloud Computing*, 2024, 13: 5. DOI:10.1186/s13677-023-005 74-9.

[11] De Keersmaeker F, Cao Y, Ndonda G K, et al. A Survey of Public IoT Datasets for Network Security Research[J]. *IEEE Communications Surveys & Tutorials*, 2023, 25(3): 1808-1840. DOI:10.1109/COMST.2023.3288942.

[12] Ahmad Z, Shahid Khan A, Wai Shiang C, et al. Network intrusion detection system: A systematic study of machine learning and deep learning approaches[J]. *Transactions on Emerging Telecommunications Technologies*, 2021, 32(1): e4150. DOI:10.1002/ett.4150.

[13] Rahman M M, Al Shakil S, Mustakim M R. A survey on intrusion detection systems in IoT networks[J]. *Cyber Security and Applications*, 2025, 3: 100082. DOI:10. 1016/j. csa. 2024. 100082.

[14] Kheddar H. Transformers and large language models for efficient intrusion detection systems: A comprehensive survey [J]. *Information Fusion*, 2025, 124: 103347. DOI:10. 1016/ j.inffus.2025.103347.

[15] Han K, Wang Y, Chen H, et al. A survey on vision transformer[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022, 45(1): 87-110.

[16] Wang H, Li G, Wang Z. Fast support vector machine classifier for large-scale problems[J]. *Information Sciences*, 2023, 642: 119136. DOI:10.1016/j.ins.2023.119136.

[17] Rainio O, Teuho J, Klén R. Evaluation metrics and statistical tests for machine learning[J]. *Scientific Reports*, 2024, 14: 6086. DOI:10.1038/s41598-024-56706-x.

[18] Al-Hawawreh M, Sitnikova E, Aboutorab N. X-IIoTID: A connectivity-agnostic and device-agnostic intrusion dataset for industrial internet of things[J]. *IEEE Internet of Things Journal*, 2022, 9(5): 3962-3977. DOI:10.1109/JIOT.2021.3102056.