Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.046237

ARTICLE

An Industrial Intrusion Detection Method Based on Hybrid Convolutional Neural Networks with Improved TCN

Zhihua Liu, Shengquan Liu[*] and Jian Zhang

College of Computer Science and Technology, Xinjiang University, Xinjiang Uygur Autonomous Regoin, Urumqi, 830000, China

*Corresponding Author: Shengquan Liu. Email: liu@xju.edu.cn

Received: 24 September 2023 Accepted: 15 November 2023 Published: 30 January 2024

ABSTRACT

Network intrusion detection systems (NIDS) based on deep learning have continued to make significant advances. However, the following challenges remain: on the one hand, simply applying only Temporal Convolutional Networks (TCNs) can lead to models that ignore the impact of network traffic features at different scales on the detection performance. On the other hand, some intrusion detection methods consider multi-scale information of traffic data, but considering only forward network traffic information can lead to deficiencies in capturing multi-scale temporal features. To address both of these issues, we propose a hybrid Convolutional Neural Network that supports a multi-output strategy (BONUS) for industrial internet intrusion detection. First, we create a multiscale Temporal Convolutional Network by stacking TCN of different scales to capture the multiscale information of network traffic. Meanwhile, we propose a bi-directional structure and dynamically set the weights to fuse the forward and backward contextual information of network traffic at each scale to enhance the model's performance in capturing the multi-scale temporal features of network traffic. In addition, we introduce a gated network for each of the two branches in the proposed method to assist the model in learning the feature representation of each branch. Extensive experiments reveal the effectiveness of the proposed approach on two publicly available traffic intrusion detection datasets named UNSW-NB15 and NSL-KDD with F1 score of 85.03% and 99.31%, respectively, which also validates the effectiveness of enhancing the model's ability to capture multi-scale temporal features of traffic data on detection performance.

KEYWORDS

Intrusion detection; industrial internet; channel spatial attention; multiscale features; dynamic fusion; multi-output learning strategy

1 Introduction

As early as 2012, General Electric Company in the United States first put forward the concept of industrial Internet [1], which breaks through the traditional industrial model and aims to enhance industrial intelligence, reduce energy consumption, and improve efficiency by connecting equipment, people and data in an open global network. In recent years, industrial Internet security incidents have occurred frequently, which not only caused severe economic losses to the entire industrial Internet industry but also caused terrible social impacts. For example, a malicious attack on the Ukrainian power system caused a massive blackout for several hours [2]. The WannaCry worm attack in May 2017 caused widespread devastation to computer networks through its rapid propagation. It infected over two hundred thousand computers quickly, significantly impacting critical institutions, including government and healthcare facilities [3]. These incidents demonstrate that increasingly sophisticated and diverse cyberattacks are infiltrating the industrial Internet sector, where security protections are weak. However, using intrusion detection systems in industrial networks can compensate for traditional network defense technologies' shortcomings and improve the entire industrial network's security system [4]. Traditional industrial internet intrusion detection systems usually use signatures or rules as indicators, and when a device or communication behavior matches predefined rules, this behavior is considered abnormal [5]. However, as more and more network attacks adopt more covert and targeted attack methods, the traditional intrusion detection system makes it difficult to meet the increasingly severe challenges posed by the actual needs of information security. Therefore, designing an accurate and efficient intrusion detection approach for industrial Internet is particularly urgent.

Artificial intelligence has become one of the most revolutionary technologies in human history. With the continuous development of AI technology, Researchers widely utilize various machine learning and deep learning methods in intrusion detection [6]. Wang et al. [7] proposed an intrusion detection approach based on Convolutional Neural Network (CNN) and LightGBM and verified the approach's effectiveness. However, since shallow machine learning methods are susceptible to spoofing attacks and are difficult to adapt to complex network attacks, more and more researchers are utilizing the adaptive and characterization capabilities of deep learning to automatically learn effective feature representations and perform pattern recognition from high-dimensional and massive network traffic data to improve the performance of intrusion detection models. Researchers have recently applied numerous deep learning techniques in intrusion detection scenarios within the industrial internet field, such as Convolutional Neural Networks (CNN [8,9]), Temporal Convolutional Networks (TCN [10]), Generative Adversarial Networks (GAN [11,12]), Long Short-Term Memory Networks (LSTM [13,14]), and some related combinations (CNN-LSTM [15]) aiming at further capturing the feature representations of network traffic.

In recent years, Wang et al. [16] and He et al. [17] proposed to introduce multi-scale information into network intrusion detection. Although these methods enhance the detection performance of the model by introducing multi-scale information, they only consider the forward information of the traffic when dealing with network traffic sequences, which leads to defects in capturing the multi-scale temporal features of network traffic and affects the detection performance of the model.

To alleviate the above problems, we propose a hybrid Convolutional Neural Network supporting multi-output strategies, called BONUS, for industrial Internet intrusion detection. Specifically, the method first utilizes a Multi-scale Convolutional Neural Network to capture multi-scale features of the network traffic data. Then, by improving the Temporal Convolutional Neural Network as a multiscale network model, we enhance the model's ability to capture multiscale temporal features, enabling the model to learn discriminative features. In addition, the two output branches of the proposed method have their gated network so that the gated network of each branch can help learn the feature representations adapted to the respective branch, further improving the model's performance in intrusion detection. Our main contributions to this study are as follows:

1. In this study, we improve the Temporal Convolutional Network into a multiscale network and construct a bi-directional structure fusing each scale's forward and backward context information for this multiscale network to enhance the model's ability to capture multiscale temporal features.

2. We present a hybrid Convolutional Neural Network approach for intrusion detection on the industrial Internet that supports multi-output strategies for anomalous traffic and intrusion category detection and utilizes features learned from the binary classification of normal and anomalous network traffic to improve the performance of the main attack category detection task.

3. Finally, we validate the proposed method on two benchmark datasets, UNSW-NB15 and NSL-KDD, and compare it with the existing optimal baseline methods. We mainly use precision, recall, and F1 scores as evaluation metrics and demonstrate the effectiveness of BONUS in industrial Internet intrusion detection.

The rest of the paper is organized as follows: Section 2 reviews related work. Section 3 describes the details of the proposed method. Section 4 describes the related datasets, configurations, and analysis of results. Section 5 summarizes and future work.

2 Related Works

Existing industrial internet intrusion detection methods can be categorized into two groups based on how they solve the problem: supervised learning-based methods and unsupervised learning-based methods. In unsupervised learning methods, the training phase involves unlabeled data to identify anomalies in new data samples, whereas supervised learning utilizes labeled training data. This section briefly reviews relevant research on intrusion detection in the industrial Internet, both supervised and unsupervised, and provides a separate overview of relevant multiscale modeling approaches.

2.1 Unsupervised Learning-Based Intrusion Detection

Intrusion detection methods based on unsupervised learning commonly rely on two techniques: clustering and AutoEncoder (AE). Clustering is a classic unsupervised intrusion detection method used to identify patterns in unlabeled data and group similar data together. It finds applications in various fields, including anomaly detection and image segmentation. AutoEncoder, on the other hand, is typically used for feature extraction, data compression, and reconstruction.

However, when the normal and abnormal samples have some common features, there is no significant difference in reconstruction error between normal and abnormal samples. At the same time, the AutoEncoder cannot extract semantic features of internal samples efficiently. To overcome the above two problems, Sun et al. [18] proposed a detection model based on mutual information maximization and mixing attention mechanism to change the effect of traditional AE on the reconstruction of internal and external samples. The model constrains the representation of the latent space by reconstructing and categorizing the rotated inner samples. The mix-and-shuffle attention mechanism makes the model pay more attention to the internal representation, and the experimental results on four datasets validate the performance of the proposed method. These techniques face significant challenges in detecting different attack classes when traffic data is unbalanced or when samples of anomalous traffic are small. To overcome this problem, Binbusayyis et al. [19] proposed an unsupervised network intrusion detection method incorporating a Convolutional Neural Network-based AutoEncoder and a class of support vector machines. The method uses only normal samples and optimizes 1D CAE for compact feature representation and OCSVM for classification to improve the model's performance. To solve the problem of significant reconstruction error and long training time in intrusion detection using Stacked Asymmetric Deep AutoEncoder, Gu et al. [20] proposed an Asymmetric Deep AutoEncoder based on the Adam optimization Algorithm. They used Random Forest to classify critical data after feature extraction. These methods have achieved better results, but unsupervised clustering methods require high data quality, and the use of unsupervised clustering methods alone may cause the problem of a high false alarm rate.

2.2 Supervised Learning-Based Intrusion Detection

In industrial internet scenarios, researchers have recently actively employed various supervised learning-based models in intrusion detection models. These models actively contribute to enhancing security and protecting against intrusions. To solve the problem of low detection rate and high false alarm rate of existing methods, Halbouni et al. [15] created a hybrid intrusion detection system model by utilizing the ability of the Convolutional Neural Network to extract spatial features and the ability of long and short-term memory networks to extract temporal features. Traditional methods use CNN and LSTM models to extract temporal-spatial features of network traffic, but previous methods do not consider the multi-feature correlation of traffic data, and to solve the above problem, Lei et al. [21] proposed a hybrid neural network model combining multi-feature correlation and temporal-spatial analysis. They used contribution-based feature selection and reconstructed multi-feature correlations between different features by constructing a triangular area map (TAM). Then, they spliced the spatial features extracted by the CNN and the temporal features extracted by LSTM to improve the model's effectiveness. To improve the ability to detect cyberattacks and to fully utilize separate models to learn the features of traffic data, Zhao et al. [10] used Dilated Causal Convolution to capture the temporal-spatial dependencies of the network traffic. They proposed an improved model based on TCN and attention mechanism, which can extract spatial and temporal features and allocate attention to the features of different attacks, improving the model's effectiveness. However, the method does not consider the impact of network traffic features at different scales on the detection performance. To address dimensional catastrophe and to achieve a balance between a low false alarm rate and a high detection rate, Mushtaq et al. [22] constructed a bipolar intrusion detection system using AE and LSTM by eliminating the noisy and less informative features and finally classifying the encoded features by using LSTM, which was analyzed on a publicly available dataset to validate the stability and efficiency of the model. To detect intrusions in a big data environment, Hassan et al. [23] proposed an intrusion detection approach based on CNN and weight-dropped LSTM for efficiently detecting network intrusions. They proved its excellent performance by comparing it to a publicly available dataset. Hand-selected feature vectors based on machine learning methods are not flexible enough to adapt to various network environments and new attack classes. In addition, the large amount of high-dimensional data increases the model training time and leads to low scalability. To address the above issues, Chen et al. [24] used a nonlinear feature extraction method of Deep Belief Networks (DBN) to extract features and ensure the original data's accuracy while reducing the original data dimensions. They used an LSTM network to obtain classification results. Experimental results show that this method has high accuracy and can spend less time updating the model with some scalability. However, these methods must consider the effect of different scale information on modeling traffic sequences, which may lead the model to ignore some significant information in the network traffic. To solve the problem of high false alarm rate caused by the unsatisfactory convergence speed and generalization ability of the Convolutional Neural Network, Wang et al. [16] proposed a Deep Multi-scale Convolutional Neural Network for network intrusion detection, which uses convolutional kernels with different scales to extract features at different levels from a large amount of high-dimensional unlabeled raw data. To solve the feature extraction problem in intrusion detection caused by large-scale high-dimensional traffic data, He et al. [17] proposed a method based on the Variational Gaussian model and one-dimensional pyramid depth-separated convolutional neural network for solving the feature extraction problem caused by large-scale high-dimensional traffic data in network intrusion detection. To solve the problem that the traditional intrusion detection algorithm cannot learn more information based on the traffic data effectively and the detection accuracy is not ideal, Kong et al. [25] proposed an intrusion detection algorithm based on the one-dimensional multi-scale residual network for industrial control systems. Firstly, the nondimensionalization of input data is realized by defining the centrosymmetric logarithmic function. Then, a one-dimensional multi-scale residual neural network model is constructed to learn the characteristic information of industrial control data, and through cross-validation, parameter tuning is realized to obtain the best model. To solve the problem that the Deep Autoencoding Gaussian Mixture Model (DAGMM) is defective in preserving the input topology, Chen et al. [26] proposed a Self-Organizing Map-assisted depth Autoencoding Gaussian Mixture Model (SOM-DAGMM), which overcomes the above drawbacks of DAGMM by well balancing the low dimensionality requirement and topology preservation requirement of Gaussian Mixture Model (GMM). In addition, Ye et al. [27] explored how to model sequence information in dynamic time scales for learning multi-scale contextual sentiment representations at different scales. This method provides new ideas for network traffic feature representation in industrial Internet.

3 Proposed Method

This section presents the general structure of the proposed intrusion detection approach for industrial Internet. Then, we will present the detailed design of its internal structure separately.

3.1 Overview of the Proposed Model

We propose a hybrid convolutional neural network method supporting multi-output strategies for industrial Internet intrusion detection, referred to as BONUS. As shown in Fig. 1, BONUS consists of four essential components: data preprocessing, attention-enhanced multiscale convolutional network, improved temporal convolutional network, and gated multi-output prediction.

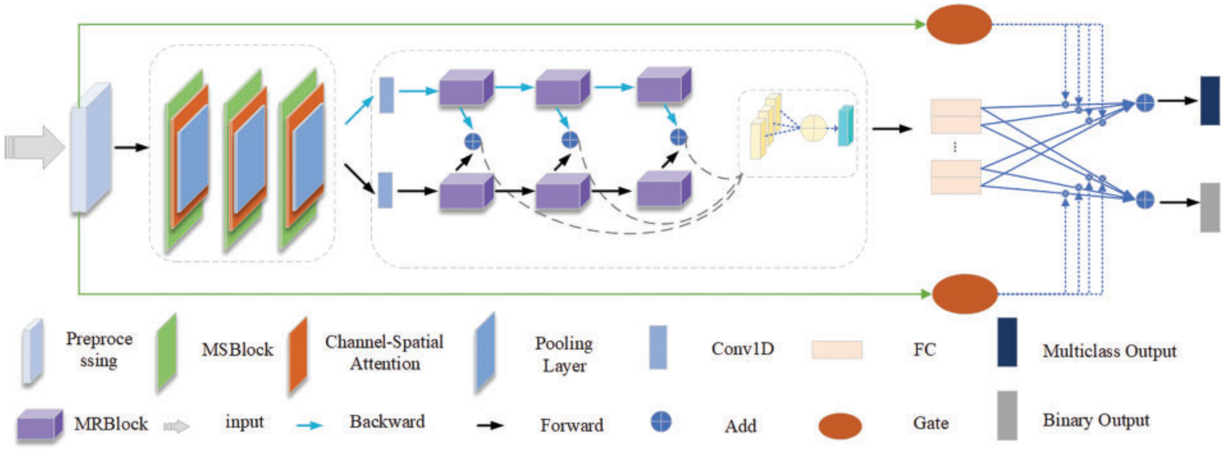


Figure 1: Framework structure of the proposed method BONUS

Specifically, the method first extracts a feature representation of network traffic data using an attention-enhanced multiscale convolutional network. Then, it models the multiscale temporal features of network traffic using an improved temporal convolutional network. BONUS has a multi-output structure. Therefore, during BONUS training, the model uses the weighted sum of binary and multi-class losses as the overall loss. The initial intention of the multi-output architecture design in the proposed method is to leverage the features learned in the binary-class branch to help the multi-class branch learn more discriminative multiscale information, thereby improving the model's performance in intrusion class detection.

3.2 Attention-Enhanced Multi-Scale Convolution Network

To alleviate the limitations of single-scale feature learning, we propose to utilize the channel space attention mechanism to help Multi-scale Convolutional Networks better capture feature representations in network traffic and name it Att-MSCNN, which consists of three tandem Multi-scale Convolutional Blocks (MSBlock). We will describe the internal structure of the MSBlock in detail in the Implementation details section. As shown in Fig. 2, the input to the Channel Spatial Attention Module is the multiscale feature $A \in R^{C*H*W}$ extracted by the Multiscale Convolutional Network.

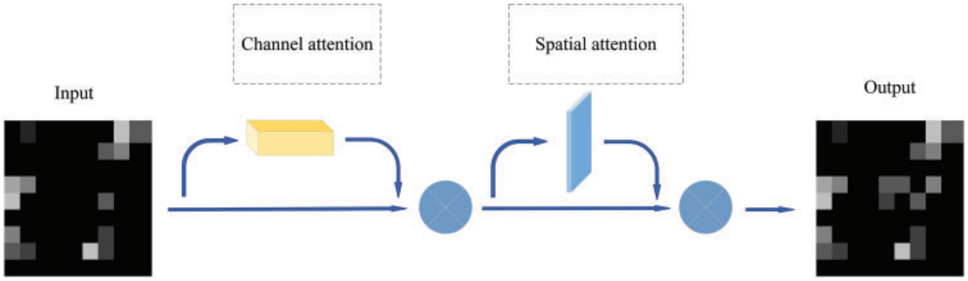


Figure 2: Channel spatial attention module

We first transform the shape of A into C ∗ N, where N = H ∗ W, (H is the height of the input, W is the width of the input, and N is the number of pixels). We obtain the feature map S by applying a Softmax layer to the product of A and its transpose, as shown in the following equation:

$$S_{ij} = \frac{e^{A_i \cdot A_i}}{\sum_i e^{A_i \cdot A_i}} \tag{1}$$

where each S_{ij} in the feature map S measures the correlation between the i_{th} channel to the j_{th} channel, and the final output E is given by:

$$E_j = \beta \sum_{i=1}^{C} \left(S_{ij} DA_{ij} \right) + A_j \tag{2}$$

According to the equation above, the final output of the channel attention is the weighted sum of all channel features and the original features. The output of the channel attention serves as the input for the spatial attention. Firstly, max pooling and average pooling operations are performed along the channel dimension, resulting in two feature maps T with shape R. Then, the feature maps T are concatenated along the channel dimension and passed through a convolutional layer to reduce the dimension to a single channel. Finally, the Sigmoid layer obtains the output of the channel spatial attention.

3.3 Improved Multi-Scale Temporal Convolutional Network

In this section, we propose an improved TCN for industrial Internet intrusion detection called ITCNet. Specifically, as shown in Fig. 1, the upper part of ITCNet requires inversion of the multiscale information of the extracted network traffic, while the lower part is a direct forward operation. Eqs. (3)

and (4), respectively, provide the outputs of the forward operation $\overrightarrow{F_j}$ and the reverse operation $\overleftarrow{B_j}$.

$$\overrightarrow{F}_{j+1} = M(\overrightarrow{F}_j) \odot \overrightarrow{F}_j \quad (3)$$

$$\overleftarrow{B}_{j+1} = M\left(\overleftarrow{B}_j\right) \odot \overleftarrow{B}_j \quad (4)$$

When j is 0, $\overrightarrow{F_0}$ and $\overleftarrow{B_0}$ are the outputs of a one-dimensional Causal Convolution from the first layer. And M denotes the MRBlock, as shown in Fig. 3, where each MRBlock consists of two Dilated Causal Convolution connected by residual connections, a sigmoid, and a spatial dropout.

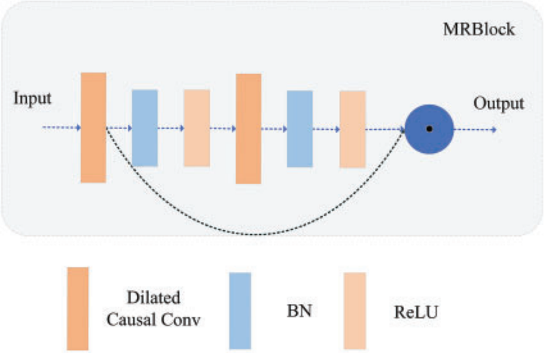


Figure 3: Internal structure of MRBlock

The role of the MRBlock is to learn the network traffic temporal feature representation A. The element-wise multiplication of the inputs and A generates multiscale information about the traffic sequence. Dilation convolution increases the receptive field without increasing the computational complexity. For two identical sub-blocks in the j_{th} MRBlock, the Dilated Causal Convolution in each MRBlock starts with a dilation rate of 2^{j-1}. The exponentially growing dilation rate allows a rapid increase in the receptive field to capture a more extended range of dependencies [27]. Also, the causal constraints ensure that the model does not leak future information.We use the add operation and the global average pooling operation to fuse forward and backward complementary contextual information about network traffic, as implemented below:

$$a_j = (Add\&GAP)\left(\overrightarrow{F_j} + \overleftarrow{B_j}\right) \quad (5)$$

Meanwhile, ITCNet avoids the need to set weights manually by dynamically assigning weights to a_j to learn the feature representation of network traffic between different scales. The model is trained through continuous iterations to achieve the optimal fusion strategy. Dynamic fusion is defined as follows:

$$R = \sum\nolimits_{j=1}^{n} W_j a_j \quad (6)$$

where the weights W_j are trainable parameters, ITCNet dynamically fuses the feature representations of different scales of network traffic to obtain a discriminative feature representation, denoted as R, of the network traffic.

3.4 Multi-Output Structure for Gated Optimization

Researchers have successfully applied the MMoE model [28] to recommendation algorithms with its underlying network called experts. Optimizing each task by sharing expert sub-models in each task while training the gated network makes the hybrid expert structure adaptable to multi-task learning. Qin et al. [29] suggested using LSTM to perform explicit representation learning from the input layer to augment the MoE layer for better sequential data processing. Inspired by the above approaches, we introduce a gated network for each of the two branches of BONUS and add a DNN structure after BONUS to form a shared expert submodel, where the gated network for each branch can learn different multi-scale feature representations to capture the relationship between the two branches, thus improving the performance of multi-class branching.

As shown on the right side of Fig. 1, BONUS has two output branches corresponding to the multi-class and binary-class outputs. Specifically, we add expert sub-models after ITCNet, where DNN structures implement the expert sub-models. The Gate network takes the multi-scale features of the network traffic captured by ITCNet as inputs. The model training process is performed with decreasing losses in both branches, allowing the two output branches to effectively utilize the expert models in different ways to capture multi-scale feature representations adapted to their respective tasks. The following equation gives the functions of binary-class and multi-class losses for BONUS:

$$\mathcal{L}_{BCE} = -\sum_{j} \left[y_j loga_j + \left(1 - y_j\right) log \left(1 - a_j\right) \right] \tag{7}$$

$$\mathcal{L}_{SCE} = -\sum\nolimits_{j} \left[y_j loga_j \right] \tag{8}$$

where y_j is the ground truth label, and a_j is the network output for a single data sample. Moreover, the total loss of our proposed method is the weighted sum of the losses of each of the two branches, formally:

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{BCE} + (1 - \alpha) \mathcal{L}_{SCE} \tag{9}$$

For the model not to focus on the performance of only one of the branch outputs, BONUS set different weights for the two outputs, with the weight parameter represented by α.

4 Results and Analysis

In this section, we conduct extensive experiments on two public datasets, NSL-KDD [30] and UNSW-NB15 [31], to evaluate the effectiveness of BONUS.

4.1 Dataset Description

NSL-KDD [30] Dataset: The NSL-KDD dataset is an improved version of the KDD-CUP dataset achieved by removing duplicate instances from the training and test sets. This dataset introduces an increased proportion of minority samples in the test set to facilitate a more comprehensive evaluation of the classification performance of diverse intrusion detection models. The training set of this dataset consists primarily of “Normal” and “DoS” samples, while the proportions of “R2L” and “U2R” samples in the dataset are significantly low. Although this dataset may not fully represent the networks existing in the real world, recent advanced research still regards it as an effective benchmark dataset, aiding researchers in comparing the effectiveness of different intrusion detection approaches.

UNSW-NB15 [31,32] dataset: The UNSW-NB15 dataset is a raw network packet collected by the Cyber Range Laboratory at the University of New South Wales, Canberra, Australia, using the

IXIA PerfectStorm tool over 31 h of monitoring. The hybrid dataset includes modern real normal activity and synthetic contemporary attack behavior extracted from network traffic monitored in 2015. The UNSW-NB15 dataset contains nine datasets from real network traffic and simulated attack datasets for 13 datasets. Each dataset contains information on source ports, destination ports, protocols, and flow identifiers. There are ten categories of data traffic classes in this dataset. In addition to the normal attacks, the UNSW dataset contains nine other attacks to simulate real network environments, including Fuzzers, Generic, Shellcode, DoS, Analysis, Exploits, Backdoor, Reconnaissance, and Worms. There are a total of 42 class-labeled features. Table 1 lists the number of samples in the training and test sets for the different categories of data in the two datasets.

Table 1: Distribution of data under different categories in the NSL-KDD and UNSW-NB15 datasets

NSL-KDD				UNSW-NB15			
Class type	Train	Test	Percentage	Class type	Train	Test	Percentage
Normal	184791	61672	51.89%	Normal	55750	18533	36.09%
Dos	32135	10695	12.87%	Generic	35432	11778	22.85%
Probe	8465	2764	32.56%	Exploits	26680	8954	17.28%
R2L	2340	779	2.52%	Fuzzers	14531	4842	9.41%
U2R	76	26	0.17%	DoS	9833	3232	6.35%
				Reconnaissance	8553	2825	5.43%
				Analysis	1646	553	1.04%
				Backdoors	1385	475	0.90%
				Shellcode	888	312	0.59%
				Worms	105	31	0.07%

To further characterize the experimental data, we describe the feature lists of the two datasets, UNSW-NB15 and NSL-KDD, as shown in Tables 2 and 3. The datasets contain three types of features: categorical, integer, and float. We use One-Hot Encoding to process the features of the categorical type into a format convenient for the deep learning model to handle.

Table 2: Description of NSL-KDD features

No.	Name	Category	No.	Name	Category
1	duration	Integer	22	is_guest_login	Integer
2	protocol_type	Categorical	23	count	Integer
3	service	Categorical	24	srv_count	Integer
4	flag	Categorical	25	serror_rate	Float
5	src_bytes	Integer	26	srv_serror_rate	Float
6	dst_bytes	Integer	27	rerror_rate	Float
7	land	Integer	28	srv_rerror_rate	Float
8	wrong_fragment	Integer	29	same_srv_rate	Float
9	urgent	Integer	30	diff_srv_rate	Float

(Continued)

Table 2 (continued)

No.	Name	Category	No.	Name	Category
10	hot	Integer	31	srv_diff_host_rate	Float
11	num_failed_logins	Integer	32	dst_host_count	Integer
12	logged_in	Integer	33	dst_host_srv_count	Integer
13	num_compromised	Integer	34	dst_host_same_srv_rate	Float
14	root_shell	Integer	35	dst_host_diff_srv_rate	Float
15	su_attempted	Integer	36	dst_host_same_src_port_rate	Float
16	num_root	Integer	37	dst_host_srv_diff_host_rate	Float
17	num_file_creations	Integer	38	dst_host_serror_rate	Float
18	num_shells	Integer	39	dst_host_srv_serror_rate	Float
19	num_access_files	Integer	40	dst_host_rerror_rate	Float
20	num_outbound_cmds	Integer	41	dst_host_srv_rerror_rate	Float
21	is_host_login	Integer			

Table 3: Description of UNSW-NB15 features

No.	Name	Category	No.	Name	Category
1	dur	Float	22	dtcpb	Integer
2	proto	Categorical	23	dwin	Integer
3	service	Categorical	24	tcprtt	Float
4	state	Categorical	25	synack	Float
5	spkts	Integer	26	ackdat	Float
6	dpkts	Integer	27	smean	Integer
7	sbytes	Integer	28	dmean	Integer
8	dbytes	Integer	29	trans_depth	Integer
9	rate	Float	30	response_body_len	Integer
10	sttl	Integer	31	ct_srv_src	Integer
11	dttl	Integer	32	ct_state_ttl	Integer
12	sload	Float	33	ct_dst_ltm	Integer
13	dload	Float	34	ct_src_dport_ltm	Integer
14	sloss	Integer	35	ct_dst_sport_ltm	Integer
15	dloss	Integer	36	ct_dst_src_ltm	Integer
16	sinpkt	Float	37	is_ftp_login	Integer
17	dinpkt	Float	38	ct_ftp_cmd	Integer
18	sjit	Float	39	ct_flw_http_mthd	Integer
19	djit	Float	40	ct_src_ltm	Integer
20	swin	Integer	41	ct_srv_dst	Integer
21	stcpb	Integer	42	is_sm_ips_ports	Integer

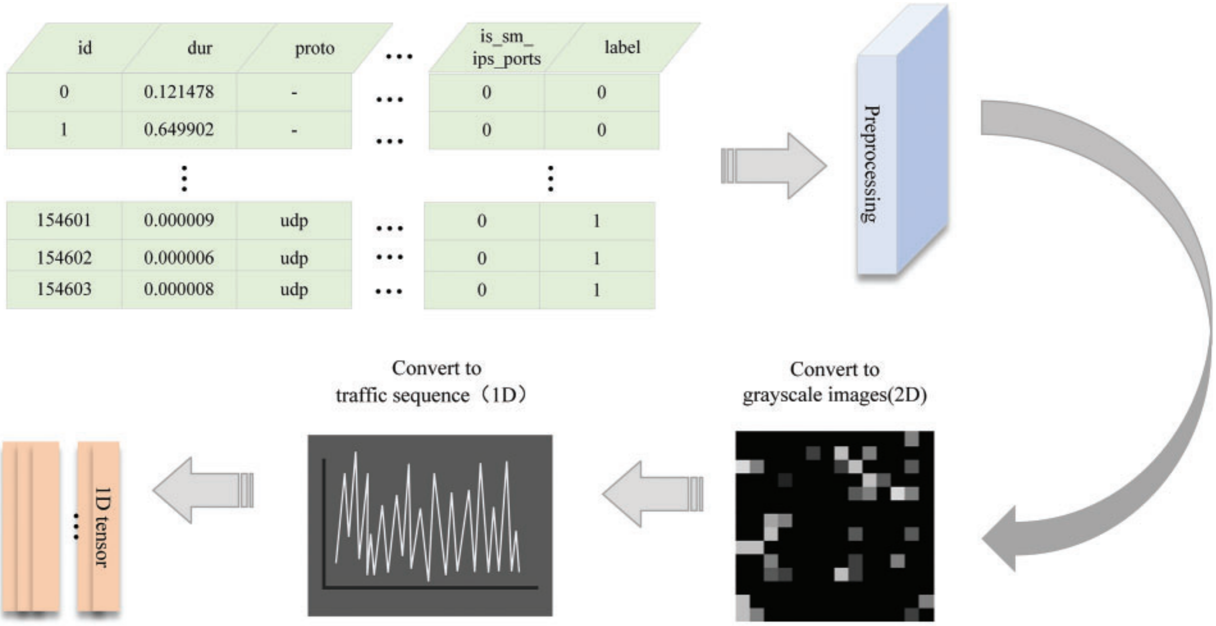
4.2 Data Preprocessing

Data preprocessing aims to optimize information collection and processing by adjusting data values in industrial Internet intrusion detection datasets. In this study, we employ two widely used datasets, UNSW-NB15 and NSL-KDD, extensively utilized in intrusion detection scenarios within the industrial Internet context.

First, we performed data cleansing to remove irrelevant records and deal with missing data. For example, in the UNSW-NB15 dataset, we remove the irrelevant "ID" column. Second, as described in Section 4.1, intrusion detection data usually contains non-numeric features, e.g., Proto and State, which are processed as numeric features using One-Hot Encoding to facilitate model processing. Finally, since there is usually a significant difference between the maximum and minimum values in the dataset, e.g., in the UNSW-NB15 dataset, the maximum value of the "sload" column is 5.27×10^9, and the minimum value is 0, to smooth the process of finding the optimal solution of the model, this paper adopts the MinMax Normalization, which is implemented by the following formula:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{10}$$

Due to the Conv2D network used in Att-MSCNN, in this paper, the input data is preprocessed into the form of a grayscale map. After Att-MSCNN processing, the transformed shape is a traffic sequence, which outputs a one-dimensional vector after the improved TCN network. Fig. 4 displays the overall data from the transformation process of our approach.



id	dur	proto	...	is_sm_ips_ports	label
0	0.121478	-	...	0	0
1	0.649902	-	...	0	0
⋮				⋮	
154601	0.000009	udp	...	0	1
154602	0.000006	udp	...	0	1
154603	0.000008	udp	...	0	1

Preprocessing
Convert to grayscale images(2D)
Convert to traffic sequence（1D）
1D tensor

Figure 4: Overall data from the transformation diagram of the proposed approach

4.3 Evaluation Metrics

To justly compare the accuracy of the proposed model with other NIDSs, we adopt accuracy (Acc), precision (Pre), recall (Rec), and F1 score (F1) as the metrics to evaluate the performance of

BONUS. Formally, these metrics are defined as:

$$\mathrm{Acc} = \frac{TP + TN}{TP + FP + FN + TN} \tag{11}$$

$$\mathrm{Pre} = \frac{TP}{TP + FP} \tag{12}$$

$$\mathrm{Rec} = \frac{TP}{TP + FN} \tag{13}$$

$$\mathrm{F1} = \frac{2 * \mathrm{Pre} * \mathrm{Rec}}{\mathrm{Pre} + \mathrm{Rec}} \tag{14}$$

where true positive rate (TP) indicates that the sample is in the positive category and predicted to be positive, true negative (TN) indicates that the sample is in the negative category and predicted to be negative, false positive (FP) indicates that the sample is in the negative category but predicted to be positive, and false negative (FN) indicates that the sample is in the positive category but predicted to be negative.

4.4 Implementation Details

In this paper, we developed BONUS using Python3 in the PyCharm-2020 development tool using Keras, an advanced neural network API integrated with TensorFlow. The input data of Att-MSCNN is the preprocessed data from Section 4.2. Att-MSCNN contains three MSBlocks (block1, block2, block3). The number of channels of block1 is {16, 32, 64}, the number of channels of block2 is {32, 64, 128}, and the number of channels of block2 is { 64, 128, 256}. Each MSBlock contains three parallel convolutional branches, where the convolutional kernel sizes are {1 ∗ 1}, {1 ∗ 1, 3 ∗ 3}, and {1 ∗ 1, 3 ∗ 3, 3 ∗ 3}, and a Channel Spatial Attention Module and a pooling layer follow each MSBlock. The padding is all set to the Same, the strides are set to 1, and the activation function is set to Relu. ITCNet contains a bi-directional structure with the lower side being forward and the upper side being backward. The first convolutional layer, Conv1D in Fig. 1, uses a convolutional kernel size of {1 ∗ 1}, and the rest of the convolutional kernel sizes are all {2 ∗ 2}. Where the dilated rate of MRBlock is set as follows: {B1:1, F1:1, B2:2, F2:2, B3:4, F3:4}. The padding is set to Casual. ITCNet splices the outputs of MRBlock with the same dilated rate during the training process and dynamically assigns weights to the spliced results before fusion. In the gated-based multi-output section, the gated network structure is added to the binary and multi-classification output structures, respectively, to help the models capture the feature representation suitable for their respective tasks. The number of expert sub-models is 8. The optimizer uses Adam with a learning rate of 0.001, and the batch size is set to 128. All the methods and models in this paper are experimented with accordingly on the following platform: the NVIDIA Tesla V100 16 GB.This research has gone through many experiments, and the ratio of the training set, validation set, and test set is 6:2:2.

4.5 Baselines

In this paper, we compare and analyze the proposed approach with the following methods on the UNSW-NB15 and NSL-KDD datasets.

(1) TACGAN-IDS [33], which proposes to construct an intrusion detection model by combining KNN and TACGAN networks and balancing the sample distribution in the NIDS by using down-sampling and up-sampling methods to improve the model detection performance.

(2) FFDNN [34], which first generates redundant minimal feature subsets based on FFSA and extracts deep features of network traffic based on feed-forward deep neural network (FFDNN).

(3) MTDL [35], which extends the self-encoder and clustering algorithms to supervised learning and proposes to develop a unified framework from three perspectives: anomaly identification, clustering, and classification to distinguish normal from attacks.

(4) IGAN-IDS [11], which proposes the use of a new imbalance-generating adversarial network, IGAN, which is utilized to generate new samples expressed in the latent space to cope with the problem of class imbalance and accordingly builds an intrusion detection system based on IGAN.

(5) ROULETTE [36], which combines attention and multi-output deep learning strategies to propose a multi-output model that can explain network intrusion detection.

(6) GMM-WGAN-IDS [12], which proposes an imbalance processing module called GMM-WGAN to extract the deep features of network traffic using the SAE module. Finally, a convolutional neural network and a long and short-term memory network are combined to detect network traffic.

(7) PyDSC-IDS [17], which proposes a combination of a variational Gaussian model and a one-dimensional pyramid depthwise separable convolutional neural network approach.

(8) IGRF-RFE [32], which proposes a hybrid feature selection method IGRF-RFE to reduce the feature dimensionality while taking into account the correlation of similar features. Experimental evidence has proved that it improves the accuracy of anomaly detection.

(9) WCGAN-XGBoost [37], which proposes to build an intrusion detection network by combining WCGAN and XGBoost, where the former is responsible for dealing with the imbalance problem, and the latter is used to classify different classes of network traffic.

(10) RL-NIDS [38] proposes network behavior modeling based on explicit and implicit feature interactions and constructs a novel network intrusion detection system, which contains both FVRL and NNRL components. Experiments prove that these two components complement each other in capturing features.

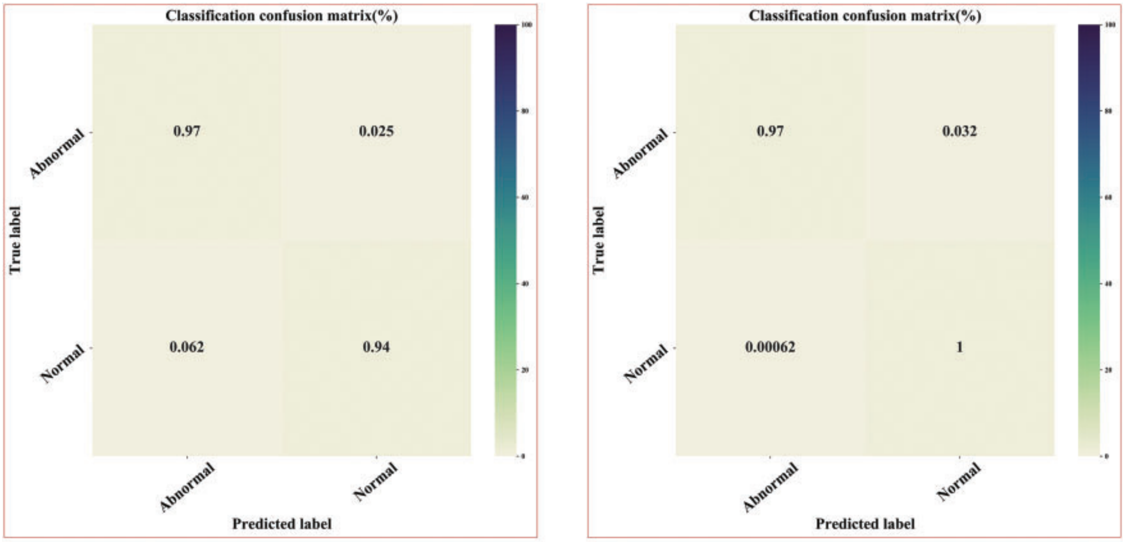
(11) CNN-LSTM [15], in which the paper creates a hybrid intrusion detection system model based on Convolutional Neural Networks and Long Short-Term Memory Networks and validates the effectiveness of the approach on three publicly available datasets.

4.6 Performance Comparison

To demonstrate the effectiveness of the proposed industrial Internet of intrusion detection model based on hybrid multiscale convolution, we compared our method with several state-of-the-art techniques on two publicly available datasets, namely UNSW-NB15 and NSL-KDD. Among these methods, BONUS achieved a multi-class F1 score of 85.03% and 99.31% on the UNSW-NB15 and NSL-KDD datasets, respectively. Its binary-class F1 score was 96.15% and 99.35%, respectively. These comparative results validate the effectiveness of our proposed method compared to other existing approaches. By comparing the metrics on both datasets, we obtained the following results.

Scenario 1: Binary classification. In the binary classification scenario, Researchers usually classify traffic data that does not belong to normal flow as either an attack or an anomaly. We present the confusion matrices for the binary classification results in both datasets in Fig. 5. Subfigure (a) in Fig. 5 depicts the confusion matrix of the proposed method on the UNSW-NB15 dataset and subfigure (b) depicts the confusion matrix of the proposed method on the NSL-KDD dataset.

Here, we focus on analyzing the commonly used metric in intrusion detection, the False Alarm Rate (FAR). FAR indicates the percentage of misclassified normal accesses as attacks. A high FAR indicates that the intrusion detection system misclassifies a significant number of normal accesses as attacks, which can impose a significant burden on the industrial Internet of Things intrusion detection system and potentially decrease system availability. Therefore, a sound intrusion detection system should have sufficient detection capability and ensure a low FAR. BONUS achieves FAR values of 3.46% and 0.27% on the two datasets, respectively. The multi-scale information of the traffic data captured by BONUS contains discriminative features that can effectively distinguish between legitimate and illegitimate requests, thus leading to a reduction in the FAR value.



(a) Confusion matrices of BONUS on UNSW-NB15

(b) Confusion matrices of BONUS on NSL-KDD

Figure 5: Confusion matrices of BONUS on UNSW-NB15 and NSL-KDD dataset

Table 4 gives the performance of BONUS and other baselines on the two benchmark datasets. On the UNSW-NB15 dataset, the F1 score of BONUS is 1.76%–3.44% higher compared to other methods, and on the NSL-KDD dataset, the F1 score is 5.25%–8.25% higher compared to other methods. Because network traffic data contains multiple levels of information, large scales can observe global trends in traffic data. In contrast, small scales can provide detailed information about traffic data. This study uses a combination of three different scales to observe traffic sequence information to capture multi-scale information that can represent traffic data more comprehensively, thus improving the expressiveness of features.

Table 4: Performance comparison in binary-class (%)

Dataset	Method	Description	Acc	Pre	Rec	F1
	CNN-LSTM	CNN, LSTM	93.78	–	–	94.77
UNSW-NB15	IGAN-IDS	FNN, IGAN, DNN	90.45	–	91.14	92.71
	TACGAN-IDS	CGAN, SGAN, infoGAN	92.39	–	94.03	94.39

(Continued)

Table 4 (continued)

Dataset	Method	Description	Acc	Pre	Rec	F1
	Ours	TCN, Gate, MSCNN, Attention	**96.16**	**96.15**	**96.16**	**96.15**
NSL-KDD	FFDNN	WFEU, FFDNN	93.2	–	–	91.1
	MTDL	TCN, Gate, MSCNN, Attention	95.5	–	–	94.1
	Ours	CNN, LSTM	**99.35**	**99.36**	**99.35**	**99.35**

However, most of the baseline above methods focus only on the fine-grained features of traffic data and only partially utilize the multi-scale information. At the same time, BONUS can capture temporal-spatial feature representations of network traffic sequences at different scales. The multi-scale information of these network flows contains discriminative features that can help improve the performance of intrusion detection models.

Scenario 2: Multi-classification. In intrusion detection in the industrial Internet, the multi-classification scenario requires a specific classification of each anomalous attack class. Figs. 6 and 7 illustrate the performance of the proposed approach on individual attack categories in the two datasets, respectively. BONUS performs well on all classes on the NSL-KDD dataset, on the UNSW-NB15 dataset in the classes Normal, DoS, Exploits, Generic, Fuzzers, Reconnaissance, and Shellcode, and relatively low performance on the classes Analysis, Backdoor Worms, this is because the distribution of this dataset is highly unbalanced and the sample imbalance of the dataset negatively affects the performance of the classifiers. In the training set, Analysis only accounts for 1.064%, Backdoor for 0.896%, and Worms for 0.068%; in the test set, Analysis only accounts for 1.073%, Backdoor for 0.922%, and Worms for 0.060%. For categories with a small number in the dataset, the training process may lead to a decrease in the accuracy and precision of the classifier. At the same time, the recall rate will also be affected.

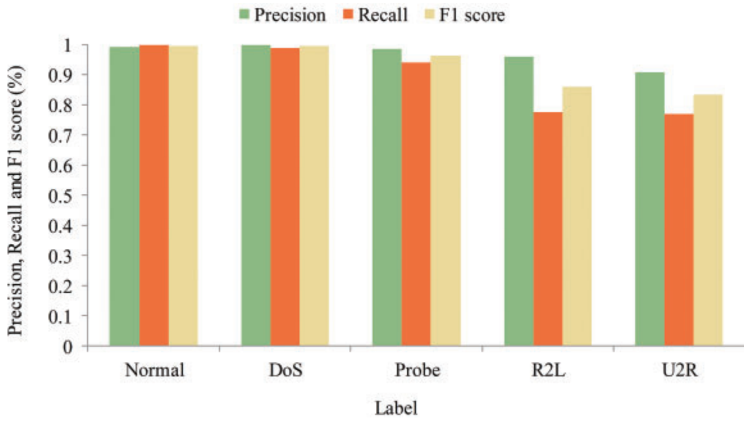


Figure 6: Performance of our approach in multi-class on NSL-KDD datasets

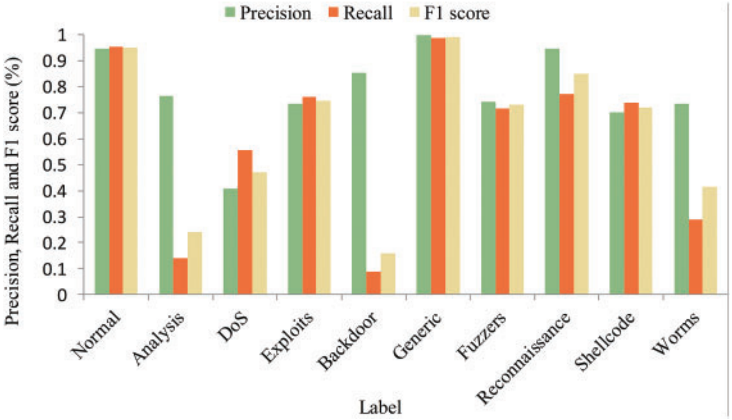


Figure 7: Performance of our approach in multi-class on UNSW-NB15 datasets

As shown in Table 5, the proposed method compares the precision, recall, and F1 score with other baseline methods on UNSW-NB15 and NSL-KDD datasets.

Table 5: Performance comparison of our approach with baselines in multi-class (%)

Dataset	Method	Description	Acc	Pre	Rec	F1
NSL-KDD	ROULETTE	Conv, Attention	81.5	–	–	79.0
	WCGAN-XGBoost	GAN, XGBoost	–	96.66	**99.65**	98.13
	PyDSC-IDS	VGM, PyDSC	81.63	83.55	–	80.63
	RL-NIDS	AutoEncoder, Triplet generator	81.37	83.69	81.38	78.79
	GMM-WGAN-IDS	SAE, CNN, LSTM, WGAN	86.59	88.55	86.59	86.88
	Ours	TCN, Gate, MSCNN, Attention	**99.32**	**99.32**	99.32	**99.31**
UNSW-NB15	ROULETTE	Conv, Attention	76.4	–	–	76.7
	WCGAN-XGBoost	GAN, XGBoost	–	81.39	81.54	81.46
	PyDSC-IDS	VGM, PyDSC	80.47	80.74	–	78.89
	IGRF-RFE	IG, RF, RFE, MLP	84.24	–	–	82.85
	CNN-LSTM	CNN, LSTM	81.83	–	–	80.87
	Ours	TCN, Gate, MSCNN, Attention	**85.13**	**86.21**	**85.13**	**85.03**

On the UNSW-NB15 dataset, the accuracy of BONUS improves by 0.89%–8.73%, and the F1 score improves by 2.18%–8.33% compared to other methods, which proves the effectiveness of BONUS in industrial Internet intrusion detection scenarios. Meanwhile, we can see that on the NSL-KDD dataset, BONUS mainly reaches the optimal level compared to other methods, which indicates that traffic features at different scales contain richer information compared to a single scale, which plays a positive role in traffic data detection.

We analyze the reasons. On the one hand, BONUS captures the diversity features of traffic data at different scales, and these diversity features can respond to the rich information of the data at multiple levels. At the same time, most other models have a single-layer architecture, i.e., they only capture the features at a single scale of the traffic data. On the other hand, the design of the bidirectional structure enables the model to integrate the complementary information of network traffic in both forward and backward directions, further improving the model's ability to capture the temporal features of traffic data. Meanwhile, the gated network of our proposed multi-output prediction part also plays a role. This part can help two different branches to learn their respective desired multi-scale features of network traffic and utilize the features learned by the binary classification branch to assist the multiclassification branch in learning more discriminative feature representations, which further improves the detection performance. However, in the recall value comparison of the NSL-KDD dataset, BONUS is 0.32% lower compared to WCGAN-XGBoost, and we analyze the reason for this. On the one hand, the XGBoost model is a kind of integrated learning algorithm based on the decision tree, which is suitable for tabular data by itself. On the other hand, using the data generation model in the WCGAN-XGBoost method alleviates the data imbalance problem. Also, it helps to improve the performance of the classifier.

In addition, in intrusion detection in industrial Internet scenarios, BONUS integrates multi-scale and forward-backward network traffic information. Compared to other methods that introduce multi-scale, they only use MSCNN to capture multi-scale traffic information. At the same time, BONUS further improves the model by improving TCN into a multi-scale model based on the former and dynamically assigning weights according to the importance of different scales so that the model can better focus on the vital temporal patterns to improve the model's ability to extract multi-scale temporal features so that BONUS can effectively extract patterns and regularities of network traffic from different scales, which helps to improve intrusion detection performance more accurately. Meanwhile, BONUS has certain flexibility and extensibility. At the same time, BONUS has a certain degree of flexibility and extensibility. ITCNet is improved from TCN and does not depend on other models to exist, and subsequent research can be further extended by introducing other model components. It can be proved through experiments that this research has practical significance in application to enhance network security and prevent production disruption and loss. By effectively detecting intrusion behaviors, industrial environments can better cope with network security challenges and ensure the reliable operation of industrial systems.

4.7 Ablation Study

In this section, we conduct a series of ablation studies to explore the effectiveness of the components of the proposed hybrid multiscale convolution-based intrusion detection approach for the industrial Internet. We use traditional CNN and LSTM methods to cross-combine with our proposed BONUS to test the contribution of the two main modules of our proposed method separately. Here, We will compare this with the following three variants:

1) CNN+ITCNet

To demonstrate the effectiveness of the multi-scale spatial features of network traffic captured by the proposed MSCNN and channel spatial attention mechanisms, we remove these two components from BONUS and use a traditional CNN network instead for comparison.

2) Att-MSCNN+LSTM

To demonstrate the effectiveness of the multi-scale information of network traffic captured by the proposed ITCNet module, we propose to use LSTM to extract the network traffic feature representation for comparison.

3) ITCNet

To demonstrate the effectiveness of the ITCNet network, we use the ITCNet module alone to capture the multi-scale information of network traffic.

Figs. 8 and 9 show that the proposed BONUS has the highest precision, recall, and F1 score on both datasets, NSL-KDD and UNSW-NB15 (Note: Since the data in Fig. 9 are close, it is especially illustrated here: the precision obtained by the Ours model is 99.32%, the recall is 99.32%, and the F1 score is 99.31%, while the precision obtained by the MSCNN+LSTM model is 99.30%, the recall is 99.31%, and the F1 score is 99.27%.) On the UNSW-NB15 dataset, the precision of BONUS is 6.57%, 4.73%, and 1.29% higher compared to the other three variants, respectively. On the NSL-KDD dataset, the accuracy of BONUS is 2.51%, 1.59%, and 0.02% higher compared to the other three variants, respectively, which indicates that the BONUS method has a low false alarm rate and can accurately identify attacks. When using recall as an evaluation metric, on the UNSW-NB15 dataset, BONUS is 6.95%, 3.08%, and 0.08% higher than the other three variants. On the NSL-KDD dataset, BONUS was 2.55%, 1.45%, and 0.01% higher than the other three variants. The experimental data show that all the essential modules in this study contributed to the effectiveness of the final model.

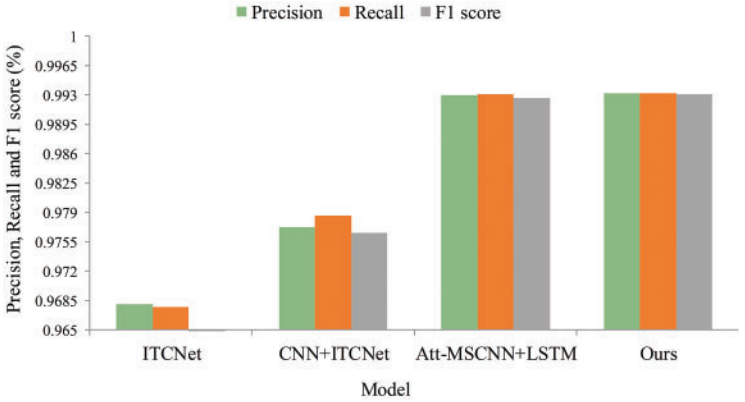


Figure 8: Performance of our approach for ablation experiments on the NSL-KDD dataset

Meanwhile, the experimental data shows that the CNN+ITCNet method reduces 4.73% on the UNSW-NB15 dataset and 2.25% on the NSL-KDD dataset compared to the whole BONUS method. The Att-MSCNN can capture more comprehensive multi-scale information of the traffic data compared to the CNN, which improves the expression of the features to improve the detection performance of the model. Meanwhile, it shows that the Att-MSCNN effectively captures the multi-scale information of network traffic. The MSCNN+LSTM method reduces 1.29% on the UNSW-NB15 dataset and 0.04% on the NSL-KDD dataset. Because the bi-directional structure in ITCNet can integrate the forward and backward information of network traffic and effectively extract the patterns and regularities of network traffic from different scales, it helps to improve the intrusion detection performance more accurately. At the same time, it also shows the effectiveness of the ITCNet method.

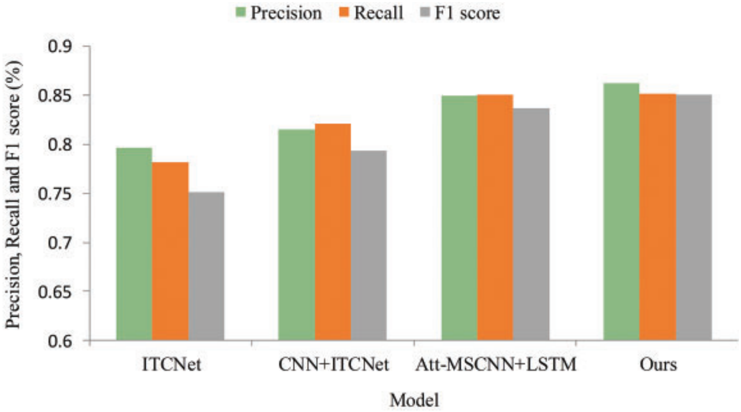


Figure 9: Performance of our approach for ablation experiments on the UNSW-NB15 dataset

Furthermore, we separately utilized ITCNet to capture the multiscale temporal information of network traffic. In the F1 score comparison on the UNSW-NB15 dataset, this method exhibited a 9.89% decrease compared to the complete BONUS method. The NSL-KDD dataset showed a 2.91% decrease compared to the BONUS method. The experimental results indicate that although ITCNet achieved a lower F1 score compared to the complete BONUS method, it still achieved good performance in terms of precision and recall. The above results suggest that ITCNet can effectively capture the multiscale temporal traffic information. Even when used independently, it can successfully detect different types of attacks, demonstrating its efficacy in practical industrial Internet intrusion detection scenarios.

As shown in Table 6, we compare the overall detection performance under different dilation rates, and our proposed method uses the bolded BONUS$_{\text{rate1,2,4}}$.

Table 6: Performances of our method on the UNSW-NB15 dataset for F1 score comparison at different scales (%)

Dilation rate	F1
BONUS$_{\text{LSTM}}$	83.61
BONUS$_{\text{rate1}}$	84.76
BONUS$_{\text{rate1,2}}$	83.95
BONUS$_{\text{rate1,2,4}}$	**85.03**
BONUS$_{\text{rate1,2,4,8}}$	84.66
BONUS$_{\text{rate1,2,4,8,16}}$	84.57

As can be seen from the comparison of the F1 score, BONUS$_{\text{rate1,2,4}}$ achieves a clear advantage, and our method improves by 0.27% to 1.42% compared to the methods using other dilation rates. Compared to the improvement of LSTM, we analyze that LSTM models sequence data through a combination of memory and gated units, where the memory unit is responsible for storing previous information, and the gated unit is used to control the flow of information to selectively ignore or retain relevant information to capture the temporal dependence of the traffic sequence. In contrast, TCN captures the feature representation of the flow sequence using a sliding convolution kernel.

However, based on the last two rows of experimental results in Table 6, it can be seen that the F1 score of BONUS gradually decreases as we add more scales, which may be because when using more significant expansion rates, samples with corresponding expansion rates do not exist in the dataset, which introduces noise that affects the model's performance. Therefore, in this paper, we used an expansion rate of {1, 2, 4} to construct the BONUS model.

The experimental data in Table 7 shows that on the UNSW-NB15 dataset, $\text{BONUS}_{\text{rate1,2,4}}$ achieves the highest recall and F1 score compared to the other scales. On the F1 score comparison, $\text{BONUS}_{\text{rate1,2,4}}$ improves by 3.39%, 3.26%, and 5.23% compared to the previous three, respectively. This further validates the effectiveness of BONUS in industrial internet intrusion detection scenarios.

Table 7: Overall performance comparison of our method on the UNSW-NB15 dataset (%)

	$\text{BONUS}_{\text{LSTM}}$	$\text{BONUS}_{\text{rate1}}$	$\text{BONUS}_{\text{rate1,2}}$	$\text{BONUS}_{\text{rate1,2,4}}$
Rec	570.31	572.56	553.38	**599.86**
Pre	778.22	739.69	**789.75**	781.85
F1	592.81	594.08	574.43	**626.67**

Meanwhile, we analyze that ITCNet is improved from TCN, which can capture traffic data information from different scales and dynamically assign weights to different scales to improve the prediction performance and generalization ability of the model. Meanwhile, the combination of ITCNet and Att-MSCNN can simultaneously extract multi-scale temporal-spatial feature information of network traffic to improve the accuracy of network traffic detection. In addition, our method has good interpretability and generalization ability and can be applied to other time series tasks.

5 Conclusions

In this paper, we propose a hybrid multiscale convolution-based intrusion detection method for industrial Internet. Specifically, we utilize an Att-MSCNN to capture multiscale traffic information. In addition, we propose ITCNet to enhance the model's ability to capture multiscale temporal features of traffic data. To improve the feature representation capability of the multiclassification branch, we introduce independent gated networks for the two output branches to simultaneously capture the multiscale feature representations adapted to their respective branches, thus improving the performance of the proposed method for industrial Internet intrusion detection. We conduct a comprehensive experimental evaluation on two publicly available datasets, UNSW-NB15 and NSL-KDD, and the experimental results show the advantages of the proposed method in classifying attacks on network traffic and improving the performance of network intrusion detection.

Despite the positive advances in network intrusion detection, In future work, we need to address some limitations that still exist in network intrusion detection: (1) Although we have improved the performance of intrusion detection by improving the model's ability to capture multi-scale temporal features, our detection model is susceptible to antagonistic attacks, where an attacker can spoof the detection system by modifying the network data. Therefore, future research must improve our method's robustness and anti-attack capability. (2) To support the research on intrusion detection in the increasingly complex industrial Internet network environment, the following research plan is to deploy the industrial Internet intrusion detection system into a distributed architecture to improve the system's scalability.

Acknowledgement: Not applicable.

Funding Statement: This work was sponsored by the Autonomous Region Key R&D Task Special (2022B01008), and the National Key R&D Program of China (SQ2022AAA010308-5).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Zhihua Liu, Shengquan Liu; analysis and interpretation of results: Zhihua Liu, Shengquan Liu and Jian Zhang; manuscript proofing: Zhihua Liu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Publicly available datasets were analyzed in this study. The UNSW-NB15 dataset from the Cyber Range Lab of UNSW Canberra is available at https://research.unsw.edu.au/projects/unsw-nb15-dataset; the NSL-KDD dataset is available at https://www.unb.ca/cic/datasets/nsl.html.

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

References

[1] P. C. Evans and M. Annunziata, "Waves of innovation and change," in *Industrial Internet: Pushing the Boundaries of Minds and Machines*, Boston, USA: General Electric, pp. 488–508, 2012.

[2] D. U. Case, *Analysis of the Cyber Attack on the Ukrainian Power Grid*. USA: Electricity Information Sharing and Analysis Centre (E-ISAC), pp. 1–29, 2016.

[3] J. Cosic, C. Schlehuber and D. Morog, "New challenges in forensic analysis in railway domain," in *Proc. of Int. Scientific Conf. on Informatics (ICIC)*, Poprad, Slovakia, pp. 61–64, 2019.

[4] W. Liang, K. C. Li, J. Long, X. Y. Kui and A. Y. Zomaya, "An industrial network intrusion detection algorithm based on multifeature data clustering optimization model," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2063–2071, 2020.

[5] D. Kus, E. Wagner, J. Pennekamp, K. Wolsing, I. B. Fink *et al.,* "A false sense of security? Revisiting the state of machine learning-based industrial intrusion detection," in *Proc. of 8th ACM Cyber-Physical System Security (CPSS '22)*, Nagasaki, Japan, pp. 73–84, 2022.

[6] I. F. Kilincer, F. Ertam and A. Sengur, "Machine learning methods for cyber security intrusion detection: Datasets and comparative study," *Computer Networks*, vol. 188, pp. 107840, 2021.

[7] Q. Wang, W. F. Zhao, X. Y. Wei, J. D. Ren, Y. Y. Gao *et al.,* "Intrusion detection algorithm based on convolutional neural network and light gradient boosting machine," *International Journal of Software Engineering and Knowledge Engineering*, vol. 32, no. 8, pp. 1229–1245, 2022.

[8] Y. L. Wang, J. H. Wang and H. L. Jin, "Network intrusion detection method based on improved CNN in Internet of Things environment," *Mobile Information Systems*, vol. 2022, pp. 1–10, 2022.

[9] G. J. Liu and J. B. Zhang, "CNID: Research of network intrusion detection based on convolutional neural network," *Discrete Dynamics in Nature and Society*, vol. 2020, pp. 1–11, 2020.

[10] P. Zhao, Z. J. Fan, Z. W. Cao and X. Li, "Intrusion detection model using temporal convolutional network blend into attention mechanism," *International Journal of Information Security and Privacy*, vol. 16, pp. 1–20, 2022.

[11] S. K. Huang and K. Lei, "IGAN-IDS: An imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks," *Ad Hoc Networks*, vol. 105, pp. 102177, 2020.

[12] J. Y. Cui, L. S. Zong, J. H. Xie and M. W. Tang, "A novel multi-module integrated intrusion detection system for high-dimensional imbalanced data," *Applied Intelligence*, vol. 53, pp. 272–288, 2023.

[13] J. W. Zhang, Y. Ling, X. B. Fu, X. K. Yang, G. Xiong *et al.,* "Model of the intrusion detection system based on the integration of spatial-temporal features," *Computers & Security*, vol. 89, pp. 101681, 2020.

[14] N. Gupta, V. Jindal and P. Bedi, "LIO-IDS: Handling class imbalance using LSTM and improved one-vs-one technique in intrusion detection system," *Computer Networks*, vol. 192, pp. 108076, 2021.

[15] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi *et al.,* "CNN-LSTM: Hybrid deep neural network for network intrusion detection system," *IEEE Access*, vol. 10, pp. 99837–99849, 2022.

[16] X. W. Wang, S. L. Yin, H. Li, J. C. Wang and L. Teng, "A network intrusion detection method based on deep multi-scale convolutional neural network," *International Journal of Wireless Information Networks*, vol. 27, pp. 503–517, 2020.

[17] J. He, X. D. Wang, Y. F. Song and Q. Xiang, "A multiscale intrusion detection system based on pyramid depthwise separable convolution neural network," *Neurocomputing*, vol. 530, pp. 48–59, 2023.

[18] L. Sun, M. He, N. B. Wang and H. B. Wang, "Improving autoencoder by mutual information maximization and shuffle attention for novelty detection," *Applied Intelligence*, vol. 53, pp. 17747–17761, 2023.

[19] A. Binbusayyis and T. Vaiyapuri, "Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and one-class SVM," *Applied Intelligence*, vol. 51, no. 10, pp. 7094–7108, 2021.

[20] Z. J. Gu, L. Y. Wang, C. B. Liu and Z. Wang, "Network intrusion detection with nonsymmetric deep autoencoding feature extraction," *Security and Communication Networks*, vol. 2021, pp. 1–11, 2021.

[21] S. W. Lei, C. H. Xia, Z. Li, X. J. Li and T. B. Wang, "HNN: A novel model to study the intrusion detection based on multi-feature correlation and temporal-spatial analysis," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 4, pp. 3257–3274, 2021.

[22] E. Mushtaq, A. Zameer, M. Umer and A. A. Abbasi, "A two-stage intrusion detection system with auto-encoder and LSTMs," *Applied Soft Computing*, vol. 121, pp. 108768, 2022.

[23] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian and G. Fortino, "A hybrid deep learning model for efficient intrusion detection in big data environment," *Information Sciences*, vol. 513, pp. 386–396, 2020.

[24] A. Chen, Y. Fu, X. Zheng and G. Lu, "An efficient network behavior anomaly detection using a hybrid DBN-LSTM network," *Computers & Security*, vol. 114, pp. 102600, 2022.

[25] D. Kong, Y. Du, M. H. Li and G. L. Li, "Industrial intrusion detection technology based on one-dimensional multi-scale residual network," in *Proc. of Int. Conf. on Computing and Artificial Intelligence*, New York, NY, USA, pp. 339–347, 2022.

[26] Y. Chen, N. Ashizawa, C. K. Yeo, N. Yanai and S. Yean, "Multi-scale self-organizing map assisted deep autoencoding gaussian mixture model for unsupervised intrusion detection," *Knowledge-Based Systems*, vol. 224, pp. 107086, 2021.

[27] J. X. Ye, X. C. Wen, X. Z. Wang, Y. Xu, Y. Luo *et al.,* "GM-TCNet: Gated multi-scale temporal convolutional network using emotion causality for speech emotion recognition," *Speech Communication*, vol. 145, pp. 21–35, 2022.

[28] J. Ma, Z. Zhao, X. Y. Yi, J. L. Chen and L. C. Hong, "Modeling task relationships in multi-task learning with multi-gate mixture-of-experts," in *Proc. of the 24th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'18)*, London, UK, pp. 1930–1939, 2018.

[29] Z. Qin, Y. C. Cheng, Z. Zhao, Z. Chen, D. Metzler *et al.,* "Multitask mixture of sequential experts for user activity streams," in *Proc. of the 26th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD'20)*, California, CA, USA, pp. 3083–3091, 2020.

[30] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. of IEEE Symp. on Computational Intelligence for Security and Defense Applications (CISDA 2009)*, Ottawa, ON, Canada, pp. 1–6, 2009.

[31] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. of Military Communications and Information Systems Conf. (MilCIS)*, Canberra, Australia, pp. 1–6, 2015.

[32] Y. H. Yin, J. L. Jang-Jaccard, W. Xu, A. Singh, J. T. Zhu *et al.,* "IGRF-RFE: A hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset," *Journal of Big Data*, vol. 10, no. 1, pp. 1–26, 2023.

[33] H. W. Ding, L. Y. Chen, L. Dong, Z. W. Fu and X. H. Cui, “Imbalanced data classification: A KNN and generative adversarial networks-based hybrid approach for intrusion detection,” *Future Generation Computer Systems*, vol. 131, pp. 240–254, 2022.

[34] S. M. Kasongo and Y. Sun, “A deep learning method with filter based feature engineering for wireless intrusion detection system,” *IEEE Access*, vol. 7, pp. 38597–38607, 2019.

[35] Q. G. Liu, D. M. Wang, Y. H. Jia, S. Y. Luo and C. R. Wang, “A multi-task based deep learning approach for intrusion detection,” *Knowledge-Based Systems*, vol. 238, pp. 107852, 2022.

[36] G. Andresini, A. Appice, F. P. Caforio, D. Malerba and G. Vessio, “ROULETTE: A neural attention multi-output model for explainable network intrusion detection,” *Expert Systems with Applications*, vol. 201, pp. 117144, 2022.

[37] V. Kumar and D. Sinha, “Synthetic attack data generation model applying generative adversarial network for intrusion detection,” *Computers & Security*, vol. 125, pp. 103054, 2023.

[38] W. Wang, S. Jian, Y. S. Tan, Q. B. Wu and C. L. Huang, “Representation learning-based network intrusion detection system by capturing explicit and implicit feature interactions,” *Computers & Security*, vol. 112, pp. 102537, 2022.