Intrusion Detection Model Using Temporal Convolutional Network Blend Into Attention Mechanism

Ping Zhao, People's Public Security University of China, China

Zhijie Fan*, Fudan University, China & Shanghai Chenrui Information Technology Company

Zhiwei Cao, Third Research Institute of Ministry of Public Security, China

Xin Li, People's Public Security University of China, China

ABSTRACT

In order to improve the ability to detect network attacks, traditional intrusion detection models often used convolutional neural networks to encode spatial information or recurrent neural networks to obtain temporal features of the data. Some models combined the two methods to extract spatio-temporal features. However, these approaches used separate models and learned features insufficiently. This paper presented an improved model based on temporal convolutional networks (TCN) and attention mechanism. The causal and dilation convolution can capture the spatio-temporal dependencies of the data. The residual blocks allow the network to transfer information in a cross-layered manner, enabling in-depth network learning. Meanwhile, attention mechanism can enhance the model's attention to the relevant anomalous features of different attacks. Finally, this paper compared models results on the KDD CUP99 and UNSW-NB15 datasets. The authors apply the model to video surveillance network attack detection scenarios. The result shows that the model has advantages in evaluation metrics.

KEYWORDS

Attention Mechanism, Causal Convolution, Cyber Security, Dilated Convolution, Network Intrusion Detection, Residual Connections, Temporal Convolutional Network, Video Surveillance Network

INTRODUCTION

With the advent of big data, the Internet constitutes an indispensable tool and platform for human society to progress, work and share information. While the network brings significant benefits to humanity, network information security also worries most network users and is widely concerning in all walks of life (Liu et al, 2018, Yin et al, 2017 and Zhang et al, 2021). Especially in the financial, medical, military, and public security fields. In these fields, abnormal network attacks and data privacy leaks have emerged, resulting in irreparable losses to the state, enterprises, and individuals (Zhang et al, 2021). So the issue of securing and maintaining a secure network environment needs to be addressed urgently.

As active defense tools, network intrusion detection models can monitor network traffic in real-time, sense hidden attacks and analyze various types of attack behaviors (Tian et al, 2021). As a result, these tools help maintain network information security and propose corresponding protection strategies. Compared with passive defense measures against network attacks, intrusion detection

DOI: 10.4018/IJISP.290832

*Corresponding Author

models can detect known attacks while discovering unknown attacks and have produced many efficient model results. With the feasibility in improving the real-time monitoring efficiency, reducing false alarm rates, and shortening detection times, intrusion detection is still an indispensable focus of research for network security defense today.

Network intrusion detection systems include techniques based on traditional machine learning, based on deep learning, reinforcement learning, and visualization learning (Wang et al, 2021). The most widely used techniques in intrusion detection include the K-Nearest Neighbor algorithm(KNN) that can reflect the difference between normal and abnormal traffic, achieving classification for various attack types without parameter estimation. Hurley et al (2016) uses principal component analysis to re-extract features and then uses KNN-based models for attack identification and classification. But large and higher-order data can make the algorithm less accurate. Compared with other machine learning algorithms, the Support Vector Machine(SVM) can improve the detection accuracy based on solving the imbalance of data samples. Teng et al (2014) and Reddy et al (2016) used SVM-based methods to effectively detect DDOS attacks, probe attacks, and other abnormal behaviors. In recent years, with the breakthroughs in deep learning research in natural language processing, image recognition, and other fields have been achieved. In contrast, the traditional machine learning methods require professionals with extensive domain knowledge to carry out manual feature extraction, as a shallow learning method has been unable to effectively cope with the massive data resources, and the network bandwidth increase caused by complex and variable data features.

Deep learning can learn the intrinsic patterns of data and adapt to high-dimensional prediction requirements, which is more efficient and has excellent potential in network intrusion detection. Recurrent Neural Network(RNN) (Manickam et al, 2017) and their variants such as Long Short-Term Memory(LSTM), Bi-directional Long Short-Term Memory(BiLSTM), Gated Recurrent Unit(GRU) (Kim et al, 2016, Roy et al and 2018, Yan et al, 2018), which take sequential data as input, can efficiently obtain the temporal information from the data. These algorithms also have deep representation, which can better achieve global detection and find potential anomalous behaviors. CNNs are feed-forward neural networks with deep structure, and their convolutional computation can efficiently and accurately achieve feature extraction (Lin et al, 2018). Liu et al (2018) use CNN combined with multiple classifiers to detect unbalanced high-dimensional data accurately. The research and use of deep learning models has become a future trend in the field of intrusion detection. But still faces problems such as training speed, computational storage, hyper-parameters adjustments, and optimization of the model due to the heavy load of the learning process. So that the implementation of network intrusion detection using deep learning methods is still a challenging problem.

This paper discussed the acquisition of spatial and temporal feature information through TCN. Then this paper introduced the Attention mechanism to give corresponding attention to the characteristics of different attack types. Finally, this paper compared the results with other deep learning models while improving the classification accuracy of different types of attacks. The main work of this paper is as follows:

(a)This paper used a network intrusion detection model based on TCN networks. It introduces the Attention mechanism, which can fuse the functions of the temporal and convolutional models in the form of layers within a single model. Moreover, the model extracts the spatial and temporal attributes of the data in parallel, and reasonably allocate attention to the characteristics of different attacks to improve the effectiveness of the model.

(b)This paper evaluated the model using a weighted average of the evaluation metrics to avoid bias in the model predictions due to data imbalance.

(c)The model was evaluated on a subset of the KDD Cup 99 10% data and UNSW-NB15.A weighted average metrics are used to evaluate the results of multiple classifications.

Related WORK

Deep learning methods have yielded good research results in the field of intrusion detection. Especially CNN and RNN models, their variants have significant advantages in capturing spatial and temporal information about the data when applied independently and in combination. The advantages of such models are briefly described next.

Multilayer Perceptron(MLP) (Ramchoun et al, 2016), also known as Artificial Neural Network(ANN), is the basis for all kinds of neural network variants. The classical MLP includes an input layer, an output layer, and an implicit layer, with full connectivity between layers. The model has a simple structure and a single internal unit function, but many units work in parallel and can learn actively to achieve efficient information processing. In addition, the structure of the MLP itself stores information on the weights of neurons to achieve a data memory function. This distributed storage also gives the MLP the ability for feature extraction and cluster analysis, so early deep learning research on network intrusion detection also uses the MLP.

Recurrent neural networks(RNN) (Zaremba et al, 2014) is a type of neural network that models sequential data. It can analyze the current output of a sequence in association with the output of the previous sequence. And the sequence possesses the ability to remember information about the data in the previous sequence. The network can then take it into account in the current computation. The hidden layers of the network have connectivity and temporal properties between them, and the output of the hidden layers includes the current and previous moment's output. Thus the RNN network has the depth to mine the temporal and semantic information in the data Expression ability (Wang et al, 2021). CHEN et al. used this algorithm to construct a classification model for wireless network intrusion detection and optimized the network structure, hyper-parameters, and generalization of the attack classification model to achieve anomalous attack detection on the network (Chen et al, 2019). However, the model's performance will degrade the longer the amount of memory data exceeds the model's load capacity when processing large data.

LSTM (Hochreiter et al, 1997) is one of the variants of RNN. RNNs in practice suffer from the gradient explosion or gradient disappearance problem, while LSTM can avoid the problem and learn the long-term temporal dependence of temporal data. LSTM has a chain-like structure with four layers of neural networks interacting to achieve long-term memory with unit states throughout the network. The forgetting unit determines which previous information is discarded, and the input unit determines the input of new information. It updates the state of the currently stored information through a state unit, and finally, the filtered information is output through the output unit. Radford et al. treat network traffic packets as words in a sequence and use word embedding to form a vector and then use LSTM to extract the temporal features of network traffic (Benjamin et al, 2018), enabling the detection of anomalous network traffic is achieved.

GRU (Cho et al, 2014) is proven to be an effective variant of LSTM with a more straightforward structure compared to LSTM. GRU replaces the input, forgetting, and output units in LSTM with the update and reset gates. The reset gate is more sensitive to short-term timing dependencies, while the update gate performs more actively for long-term dependencies. The GRU is not only able to circumvent the gradient disappearance problem in back-propagation but also able to outperform the LSTM in terms of computational speed and efficiency (Cho et al, 2014). The intrusion detection model in the literature (Li et al, 2021) uses two GRU structures to store data to achieve non-linear classification decisions. The use of GRU allows the model to obtain the best detection performance using the smallest sample size, which converges faster and reflects the efficiency of GRU.

CNN is a feed-forward neural network with a deep structure, and a typical CNN consists of the input layer, the convolutional layer, the pooling layer, and the fully connected layer. The use of convolutional computation can extract data space features accurately and efficiently (Xiao et al, 2019). Together with its advantages of fewer network parameters and translation invariance, some scholars to construct models for anomaly attack detection by converting network traffic into the form of images in the network intrusion detection problem (Liu et al, 2020). In addition, CNN is also good at handling data with statistically smooth and locally correlated properties, so they are also used to perform the selection of traffic features and set the cost function weights for attack classification based on the number to solve the data imbalance problem in intrusion detection data (Naseer et al, 2018). However, CNN cannot learn serial correlation and cannot understand long-term data dependencies (Liu et al, 2019).

CNN-LSTM model, CNN, and LSTM models can acquire temporal and spatial characteristics of data, respectively. Many scholars have combined the advantages of the two types of models to construct intrusion detection models based on CNN and LSTM (Liu et al, 2019 and Yao et al, 2021). Parallel local features of attribute information are extracted through convolution and pooling operations of CNN networks. The LSTM is then used to capture long-time dependent features, fully considering the interactions between feature information. The fusion model helps to reduce the false alarm rate of model detection while further achieving the improvement of intrusion detection model performance and detection accuracy.

The above methods are a combination of two independent models encoding the spatial and temporal characteristics of the data. The network structure is highly complex, involves more parameters, yet there is still room for improvement in the detection time of intrusion detection models. Lea et al. pioneered TCN for segmenting the actions of people in videos (Lea et al, 2016). TCN is considered an optimization of the CNN network, which is a typical one-dimensional convolutional neural network (Fan et al, 2021). Furthermore, TCN subsequently achieved optimal results in areas such as weather prediction (Yan et al, 2020), sound event localization and detection (Guirguis et al, 2020), probabilistic prediction (Ngo et al, 2021), and machine translation (Kalchbrenner et al, 2016). Compared with the networks mentioned above, the TCN network has a clear and concise structure, can be parallelized, and provides more accurate results. In this paper, the model will be migrated to applications related to network intrusion detection based on its characteristics. On the other hand, the attention mechanism has also received much attention in recent years. The attention mechanism imitates the way humans observe objects and discover essential information from a large amount of information (Shu et al, 2020). The attention mechanism is also introduced into intrusion detection, which can better learn attack features.

METHOD

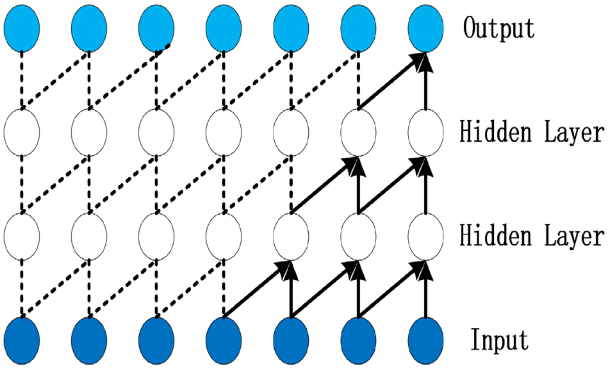
Temporal Convolutional Network

TCN (Bai et al, 2019) is composed of a one-dimensional fully convolutional layer with the same input and output lengths of causal convolution, dilated convolution, and a residual module. TCN uses zero paddings to keep the lengths between layers the same. The causal convolution in the network allows layers to form causal relationships, thus avoiding information leakage. Meanwhile the residual module and dilated convolution control over the length of memory. They ensure the model has long-term memory capability and the gradient disappearance problem is avoided, further enhancing the model's predictive capability (Zhai, 2021 and Yao et al, 2021). Firstly, TCN inherits the characteristics of the receptive field in convolutional neural networks with a flexible receptive field, which can be adjusted to the size of the receptive field according to different task characteristics. Secondly, the sharing of parameters in different stages allows the model to avoid the gradient vanishing problem in the recurrent neural network and enables parallel processing of temporal data without processing the data in a specific order. Finally, the sharing of convolutional kernels allows for a lower memory footprint and a shorter network feedback loop, making the model faster to train and validate.

Causal Convolution

Causal convolution was first proposed in WaveNet. Causal convolution has a unidirectional structure, i.e., it can only perceive memory history information with strict temporal constraints. The structure is shown in Figure 1. For the values at the moment T of the previous layer, the causal convolution depends only on the next layer at the moment T and its historical values, i.e., the elements in the output sequence, depend only on the input sequence elements and the historical elements. To ensure that the inputs and outputs have the same length, they are padded with zeros. Zero paddings are applied to the front end of the input data sequence to ensure causality in the convolutional layer. In the absence of expansion, the total amount of padding required to maintain the same length as the input is kernel_size^{-1}.

Figure 1. Causal convolution



Dilated Convolution

Traditional convolutional neural networks use pooling operations to maintain features and avoid overfitting. But when the convolutional layers deepen, the network requires more parameters and the computational complexity increases. Dilated convolution (Yu et al, 2016) can cope with this problem well. The structure is shown in Figure 2. Dilated convolution changes the size of the convolutional kernel by adding a dilated rate to the standard convolution, which represents the number of intervals in the kernel, as shown in Figure 3. Compared to traditional convolutional networks, dilated convolution can achieve a larger receptive field with the same number of layers, and the larger the receptive field, the better the data memory capability.

Figure 2. Dilated convolution

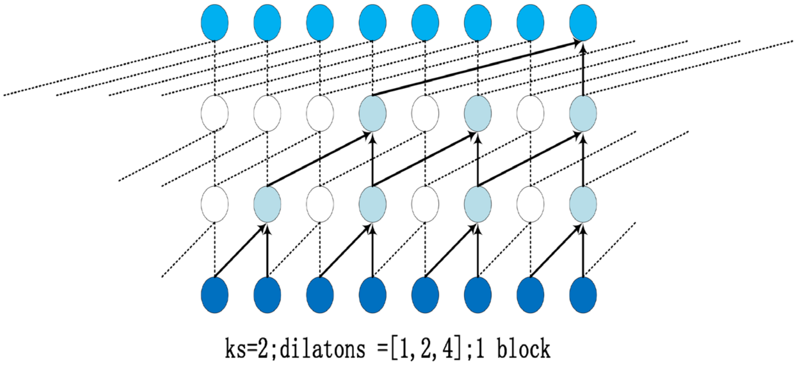
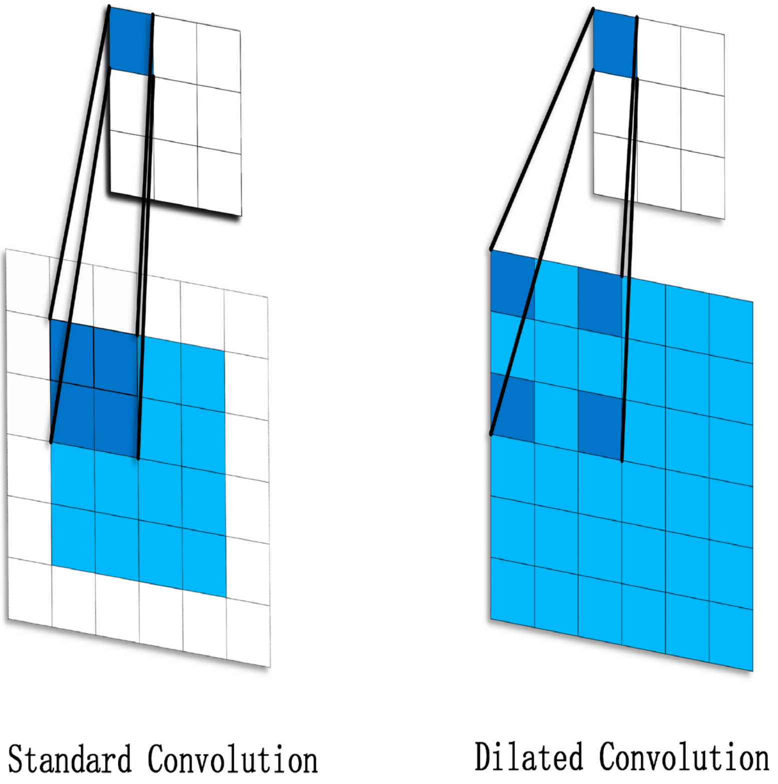


Figure 3. Expansion of the receptive field



The equation for the dilated convolution is as follows:

$$F(s)= (x *_{d} f)(s) = \sum_{i=0}^{k-1} f(i)x_{s-d\cdot i} \quad (1)$$

In the formula, d represents the dilated rate, and k is the size of the convolution kernel. When d is equal to 1, the dilated convolution is transformed into a standard convolution. The control of the receptive field range is achieved by changing the value of d. The receptive field in a convolutional neural network refers to the size of the area mapped by the nodes in the feature map on the input map. Expanding the receptive field upgrades the long-term memory capability. The receptive field is linearly related to the size of the convolutional kernel and the number of convolutional layers. The formula for calculating the receptive field size is as follows:

$$R_{field}= 1+2\cdot(K_{size}-1)\cdot N_{stack}\cdot\sum_{i} d_{i} \quad (2)$$

Where N_{s} is the number of stacks, N_{b} is the number of residual blocks per stack, d is a vector containing the dilations of each residual block in each stack, and K is the kernel size. Optimization of the dilated convolution is achieved by increasing the range of the receptive field, but multiple layers of stacked dilated convolution layers can also lead to the problem that some of the data is not involved in the computation. To ensure adequate access to information when deep convolutional neural networks are constructed, the dilated rate needs to be increased exponentially by a factor of 2 as the depth of the network increases.

Residual Connections

As the depth of the network increases and the number of network layers increases, the more abstract and meaningful the features extracted become. Nonetheless, simply stacking the network layers would lead to a gradient problem, which would be solved using regularization and initializing the weight parameters. Yet, network degradation would arise. The idea of a residual network allows for a constant mapping of the redundant layers of the network, making the deeper network equivalent to a shallow network. It replaces one layer of convolution operations with a residual module, thus solving the network degradation problem. The residual module enables the network to transfer information in a cross-layer connection. The structure of the residual module is shown in Figure 4. The module provides two types of connections: identify mapping (shortcut) and residual mapping, with the residual mapping being set to 0 as the network reaches optimality and continues to deepen the network. The TCN residual module is internally connected as a residual function from the input after two rounds of dilated convolution, weight normalization, Relu activation, and Dropout. If the input undergoes one-dimensional convolution filters, it is connected as a shortcut. The introduction of the residual module in the TCN enables in-depth training of the data while avoiding the problems associated with multi-layer networks.

Attention Mechanism

Attention mechanisms were first applied to computer vision, where the human eye focuses on a specific part of a target object to catch the critical information when looking at a picture or an object (Liu et al, 2021). Attention mechanisms can be applied in various fields such as machine translation and sentiment classification. The key principle is to use attention to determine which part of the input data is of most interest. Then extract features from the key part and use the important information to complete the subsequent classification task. The attention mechanism helps to improve the interpretability of neural network models and reduce some of the drawbacks associated with stacking deep neural networks. In this paper, the attention mechanism is applied to an intrusion detection model. For input multi-feature data, the attention mechanism can assign important weights to different inputs. It focuses on the feature content most relevant to anomalous attacks and ignore the noise and redundant information in the input, further helping the model to improve the classification effect.

Intrusion Detection Model Based on TCN- Attention Mechanism

In this paper, the data is pre-processed and fed into a TCN-Attention model to further extract features from the data. The model shows more optimized results on the multi-classification problem for a 10% subset of the KDD CUP99 and UNSW-NB15 datasets. The model has also demonstrated its ability to detect accurately in practical applications. The model structure is shown in Figure 5.

The model consists of five parts. The input data of the model is sent to the TCN network layer in shape as (timesteps, input_dim) after numerical and normalization processing. At this stage, a wider receptive field enables better memory and learning of the temporal and spatial information of the input data. Followed by the application of the attention mechanism to further deepen the focus on and attack features. It learns the differences between various attacks, and improves the model's ability to discriminate between the features of different kinds of attacks. Finally, a fully connected layer is used to pass the output values to the Softmax function to complete the multi-classification intrusion detection task. The causal convolution, inflation convolution, and residual modules in the TCN layer widen the receptive field and extract the Spatio-temporal characteristics of the data in parallel. While the input data for intrusion detection are all high-dimensional and contain features of different attack types. The attention mechanism introduced by the model identifies normal and

Figure 4. Residual connections

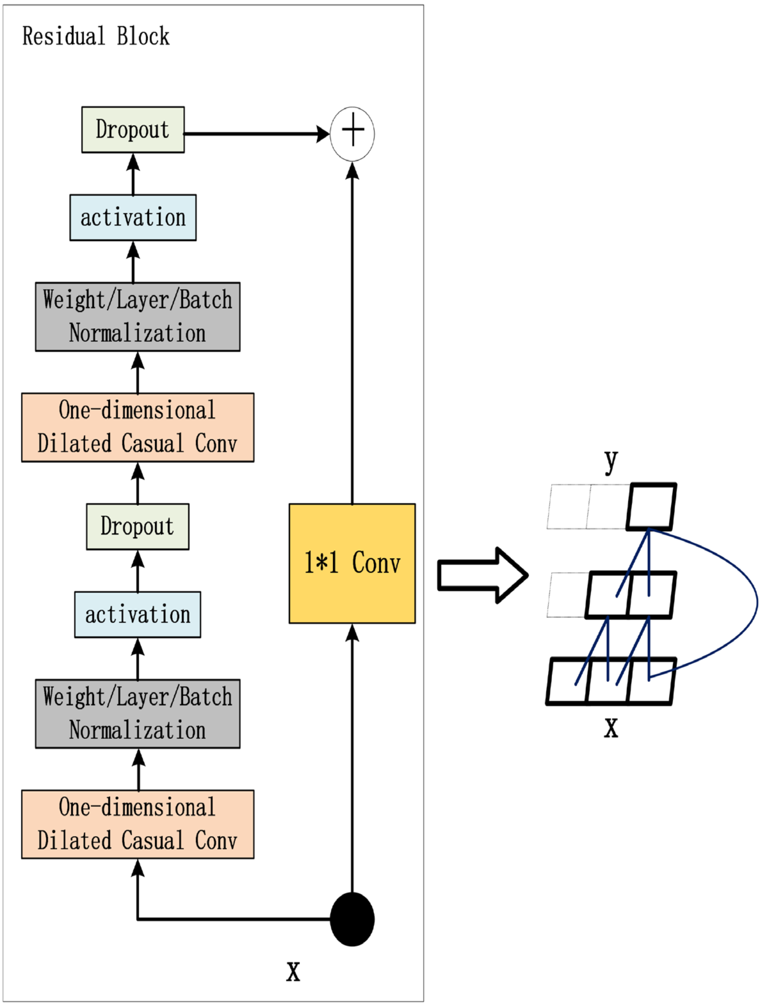
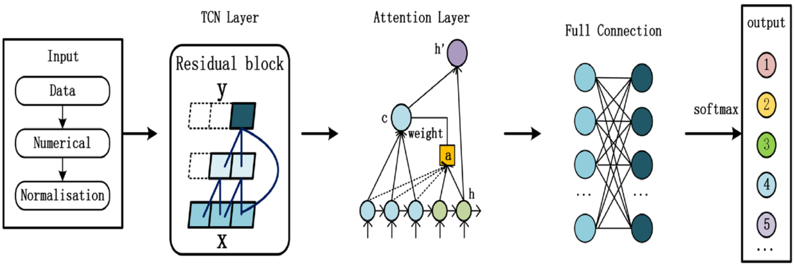

Residual Block
Dropout
activation
Weight/Layer/Batch Normalization
One-dimensional Dilated Casual Conv
Dropout
activation
Weight/Layer/Batch Normalization
One-dimensional Dilated Casual Conv
1*1 Conv
X
y
X

Figure 5. Model structure



abnormal data using the information most relevant to the abnormal attack. Besides, it also focuses on the differences in characteristics exhibited by the various forms of attack to make better judgments and solve the multi-classification problem. Finally, to ensure model training efficiency, accuracy and to prevent overfitting and make the model more robust, we set dropout=0.5 and introduce layer normalization to normalize all neuron nodes in each layer for a single sample. The TCN-Attention model combines the advantages of the temporal model, the convolutional model, and the attention mechanism, resulting in a simpler and more efficient internal structure.

EXPERIMENTS

Experiment Environment

The experiment environments include Tensorflow (2.4.1), Keras (2.4.3), Skit-learn, Python (3.7), and can be run on various platforms, such as CPU and GPU. Moreover, this paper introduce Adam optimizer and loss functions based on the mean-square error in training. The details are shown in Table 1.

Table 1. Experiment environment

Project	Environment
Operating System	Window 10
GPU	Nvidia 1660Ti
VRAM	6G
Memory	16G
CPU	Xeon Gold 6240
Function Base	Tensorflow (2.4.1) & Keras (2.4.3)
Programming Environment	Python(3.7)

Datasets

KDD CUP99

KDD CUP99 is a pre-processed extraction of approximately 5 million records of network connections based on the DARPA 98 dataset and is widely used for intrusion detection and assessment. The data set contains four main categories of 39 attack types, of which 22 attack types appear in the training set, and another 17 unknown attack types appear in the test set. The four types of abnormal attacks are DOS, R2L, U2R, and Probe.

The dataset uses 41 features to describe the network state, with item 42 being the corresponding label. Due to a large amount of duplicated and redundant data in the original dataset, a subset of 10% of the KDD CUP99 dataset was used for training and evaluation of the model, containing a total of 494,021 connection records, and the ‘corrected’ sample was selected as the test set, which contained the types of attacks that did not appear in training. The specific details of this data subset are shown in Table 2.

UNSW-NB15

The raw network packets of the UNSW-NB15 dataset was created by the IXIA PerfectStorm tool in the Cyber Range Lab of UNSW Canberra for generating a hybrid of real modern normal activities

Table 2. Details of KDD CUP99 10% subset

Types	KDD Cup 99 10% dataset	corrected
Normal	97,278	60,593
DOS	391,458	229,853
Probe	4,107	4,166
R2l	1,126	16,189
U2r	52	228
Total	494,021	311,029

and synthetic contemporary attack behaviors. The tcpdump tool was utilized to capture 100 GB of the raw traffic. The Argus, Bro-IDS tools are used and twelve algorithms are developed to generate features with class label. The nine types of attacks in this dataset are Analysis, Backdoor, Dos, Exploits, Fuzzers, Generic, Reconnaissance, Shellcode and Worms. The training set and testing set of the data set are used in the experiment, the number of records in the training set is 175,341 records and the testing set is 82,332. There are 44 features contained in the two datasets. The more details of attacks in the data set are shown in the table 3.

Table 3. Details of the UNSW-NB15

Types	Training set	Testing set
Normal	56000	37000
Analysis	2000	677
Backdoor	1746	583
Dos	12264	4089
Exploits	33393	11132
Fuzzers	18184	6062
Generic	40000	18871
Reconnaissance	10491	3496
Shellcode	1133	378
Worms	130	44
Total	175341	82332

Data Pre-Processing

Since neural networks operate on numeric data, this paper need to encode the categorical columns. Protocol type, service, connection status (flag), and 22 specific attack types in KDD dataset and proto, state, service, attack_cat in UNSW-NB15 dataset are all identified by text, the text corresponding to these features needs to be mapped to a numerical representation. Then deletd the rows with missing values. Next, for discrete feature in the data set, One-hot encoding is performed to take the discrete feature values to correspond to points in the Euclidean space for better model learning. One Hot encoding allows us to convert each category of a categorical feature into its own feature. Moreover, Neural Networks are sensitive to data with features that have large differences in their numeric range, for continuous feature values, normalization is performed in order to ensure that all values in every numeric column are between 0 and 1. This is important in ensuring that no features are overshadowed by others during the NN learning process. The formula of normalization is as follows:

$$x^{'} = \frac{x\text{-}x_{\min}}{x_{\max} - x_{\min}} \tag{3}$$

Four types of attack labels and normal labels exist in the KDD CUP99 data set, these five categories are labeled with one-hot encodes, as shown in Table 4.

In the UNSW-NB15 data set, there are nine categories of attacks. These categories are labeled with one-hot encodes, as shown in Tables 5.

Table 4. One-hot encode for labels (KDD CUP99)

Number	Types	One-hot encode
0	Normal	[1,0,0,0,0]
1	Dos	[0,1,0,0,0]
2	Probing	[0,0,1,0,0]
3	R2l	[0,0,0,1,0]
4	U2r	[0,0,0,0,1]

Table 5. One-hot encode for labels (UNSW-NB15)

Number	Types	One-hot encode
0	Normal	[1,0,0,0,0,0,0,0,0,0]
1	Analysis	[0,1,0,0,0,0,0,0,0,0]
2	Backdoor	[0,0,1,0,0,0,0,0,0,0]
3	Dos	[0,0,0,1,0,0,0,0,0,0]
4	Exploits	[0,0,0,0,1,0,0,0,0,0]
5	Fuzzers	[0,0,0,0,0,1,0,0,0,0]
6	Generic	[0,0,0,0,0,0,1,0,0,0]
7	Reconnaissance	[0,0,0,0,0,0,0,1,0,0]
8	Shellcode	[0,0,0,0,0,0,0,0,1,0]
9	Worms	[0,0,0,0,0,0,0,0,0,1]

Evaluation Metrics

In this paper, as a classification task, the model will be evaluated using accuracy, precision, recall, and F1-Score when evaluating the performance of the intrusion detection model. Accuracy is the ratio of the total number of correct classifications to the classifications in the task. The formula for calculating the accuracy is as follows:

$$\text{Accuracy}= \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{4}$$

The calculation of precision, Recall, and F1-Score need to be based on the four categories of the model's final prediction results, as shown in Table 6.

Table 6. Confusion Matrix

Real Tags	Predictive Tags	
	Attack	**Normal**
Attack	True Positive(TP)	False Positive(FN)
Normal	False Negative(FP)	True Negative(TN)

True Positive means that attack data is correctly classified as an attack by the model. False Positive means that normal data is incorrectly classified as an attack by the model. False Negative means that attack data is incorrectly classified as normal by the model. True Negative means that normal behavior is correctly classified as normal by the model. Precision refers to the proportion of the sample with an optimistic prediction that is a true positive sample. Recall is the proportion of positive cases in the sample that are predicted correctly. F1-Score is the summed mean of precision and Recall, reflecting the stability of the model. These three evaluation indicators are calculated as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{5}$$

$$\text{Recall}= \frac{\text{TP}}{\text{TP+FN}} \tag{6}$$

$$\text{F1} - \text{Score} = \frac{\text{2Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

Considering the extreme imbalance in the number of different sample categories in the intrusion detection data set, this paper uses the weighted-average approach to calculate these three evaluation metrics: Wa-precision, Wa-recall, Wa-F1score with the following formula, a_i denotes the weight occupied by category i:

$$\text{Wa-Precision}= \sum_{i=1}^{n} \alpha_i * \text{Precision}_i \tag{8}$$

$$\text{Wa-Recall}= \sum_{i=1}^{n} \alpha_i * \text{Recall}_i \quad (9)$$

$$\text{Wa-F1}= \sum_{i=1}^{n} \alpha_i * \text{F1-Score}_i \quad (10)$$

Analysis of Experimental Results

The TCN-Attention network intrusion detection model designed in this paper combines the advantages of extracting temporal data features and capturing spatial characteristics of data in one model. In addition, attention mechanisms are introduced to suppress the negative effects of irrelevant information on the effectiveness of intrusion detection and to focus on the relevant characteristics of the attack data. The model is used for parallel extraction of spatio-temporal information from the KDD CUP99 and UNSW-NB15 datasets. While the TCN allows for long term memory of the data and the attention mechanism captures the anomalous features of the attack data, the combination of which further improves the accuracy of the network anomaly attack detection. A weighted average metric is used to evaluate the results of multiple classifications, taking into account the extreme imbalance in the

Table 7. Comparison of weighted average evaluation metrics for all models (KDD CUP99)

Baseline	Wa-precision	Wa-recall	Wa-F1 score	Accuracy
TCN-Attention	**94.01%**	**92.33%**	90.20%	**92.81%**
TCN	93.75%	90.78%	89.96%	92.38%
CNN-LSTM	91.68%	91.47%	89.49%	92.01%
CNN	93.57%	92.23%	**92.20%**	92.12%
LSTM	91.68%	91.47%	89.49%	91.29%
MLP	86.23%	77.50%	80.51%	85.68%
RNN	90.86%	91.96%	89.91%	90.03%
GRU	91.92%	92.01%	89.98%	92.00%

Table 8. Comparison of weighted average evaluation metrics for all models (UNSW-NB15)

Baseline	Wa-precision	Wa-recall	Wa-F1 score	Accuracy
TCN-Attention	**72.39%**	**72.92%**	**70.55%**	**72.92%**
TCN	63.71%	70.41%	62.89%	70.98%
CNN-LSTM	55.34%	68.37%	58.90%	68.38%
CNN	59.83%	72.38%	62.35%	70.27%
LSTM	49.15%	61.65%	53.14%	61.65%
MLP	55.65%	40.38%	30.29%	40.30%
RNN	64.39%	68.82%	69.64%	68.02%
GRU	58.04%	68.75%	61.68%	68.75%

number of attack samples. Table 7, table 8 compares the evaluation metrics of the TCN-Attention model and the classical model of network intrusion detection models that can extract temporal and spatial information. In the KDD CUP99 dataset, the TCN-Attention model significantly outperformed the other models in Wa-precision, Wa-recall, and accuracy metrics, and ranked second in Wa-F1 score metric results. In the UNSW-NB15 dataset, the TCN-Attention model outperformed the other baselines in all metrics and showed the best classification results.

Figure 6. Comparison of results (KDD CUP99)

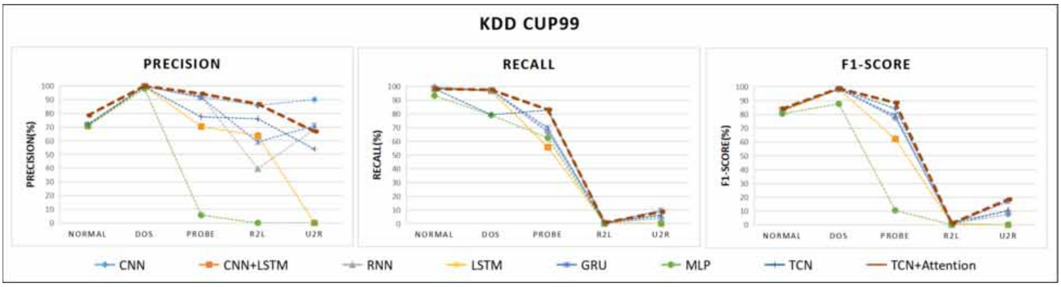


Figure 7. Comparison of results(UNSW-NB15)

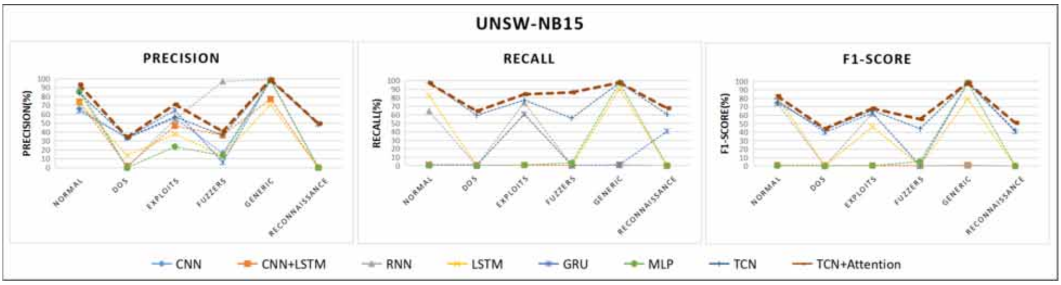


Figure 6, figure 7 show the results of multi-classification tasks. For the KDD CUP99 dataset, the TCN-Attention model (red line) outperforms other models in more than 70% of the evaluation metrics, in which the proposed model outperforms other models in all metrics for both “Dos” and “Probe” types of attacks. The legend shows that the model does not detect well in U2r and R21. The underlying reason for this is some of the attacks are sparsely sampled, so the model cannot learn the two anomalous attacks attributes well, therefore underperforms. F1-score is the summed average of the precision and recall rates, which is a comprehensive assessment of the model’s performance. The overall performance of the TCN-Attention model is the best. For the UNSW-NB15 dataset, the TCN-Attention model outperforms other models in more than 85% of the evaluation metrics, the six labels “Normal”, “Dos”, “Exploits”, “Fuzzers”, “Generic”, “Reconnaissance” have more recorded data and the model has better detection results, so this paper only show the results of these six types in the figure 7. While for the labels “Analysis”, “Backdoor”, “Shellcode”, “Worms”, all models are unable to learn these attacks features well for detection. Due to the imbalance of the data and the low number of anomalous records in the dataset.

Comparing the results of the evaluation metrics in the above tables, the authors find that the MLP model has the worst results. The MLP is a feed-forward neural network and is not good at handling datasets with high-dimensional features. Furthermore, the model also does not have long-term memory capability and has poor access to information about the features, so the model is not effective at multi-classification. The three temporal models, RNN, LSTM, and GRU are more effective than MLP because they can capture the temporal information in the traffic data, which further enhances the effectiveness of the models. The CNN-LSTM model incorporates a convolutional layer capable of extracting spatial elements on top of extracting temporal attributes of the data. But the direct stitching of the two models is less effective than using the CNN model alone. Possible reasons for this result are that the CNN and LSTM models are incompatible in terms of connectivity, which leads to some feature attributes being ignored when passing information between the two networks during the extraction of features. Secondly, the spatial knowledge learned by CNN and the temporal knowledge learned by LSTM are not well integrated. The information obtained is not fully communicated with each other, resulting in a lack of significant improvement in classification. The CNN model is second only to the TCN model and the TCN-Attention model, indicating that the convolutional and pooling layers are better able to extract feature attributes. Nevertheless, the model cannot extract time-series information and still has an insufficient understanding of the data, leaving some room for improvement. The TCN model is able to learn the spatio-temporal properties of the data well. But, without the Attention mechanism, it does not capture the features of the anomalous attack data deeply enough.

One of the advantages of the TCN-Attention model is the simpler internal structure of the model. The ability to combine the advantages of a convolutional neural network with the advantages of a temporal model to extract the spatio-temporal characteristics of the data in a single model. In addition, the use of TCN avoids the gradient problem, enables long-term memory of the data. It improves the shortcomings of the convolutional model and fully explores the content of the data. The second advantage of the model lies in the introduction of the attention mechanism, which increases the interpretability of the neural network model. The attention mechanism focuses on the relevant features of different attack types for the input high-dimensional data, learns the data selectively. And again, it excludes the interference of irrelevant factors, allowing the model to have a deeper understanding of the data and to mine the information more fully. Therefore, the model is able to obtain better results when applied to detect anomalous attacks in practical scenarios.

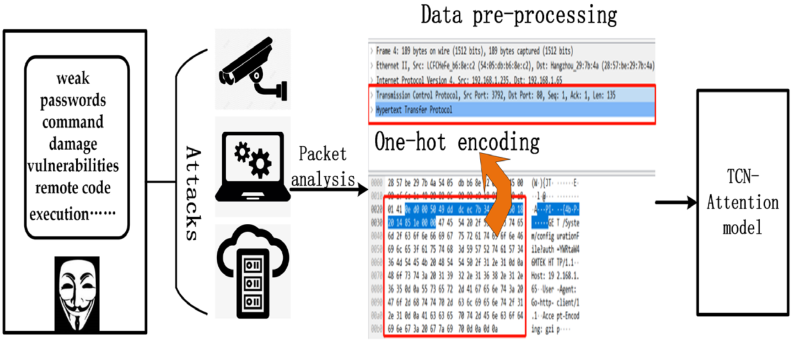
APPLICATION

Video Surveillance Network Data

First, this paper researched the front-end devices deployed in the video surveillance network. Moreover, authors looked up vulnerability information and attack scripts for different brands of device models. Then, the authors studied the video surveillance network topology and business application scenarios so that the authors can understand the flaws in the network deployment. Finally, the attacks on the video surveillance network were attempted using various artifices to understand the characteristics of the attack methods. After completing the above preparatory work, the authors found that attacks using weak passwords, command damage, device and system vulnerabilities, and remote code execution were able to successfully bypass network defenses and achieve control of the network. In addition, these attacks have a higher success rate, are simple to implement, and have a greater negative impact on the network. After mastering the weaknesses of the video surveillance network, this paper conducted a network attack and defense maneuver at a video surveillance network in an area of Beijing. To allow for a more diverse sample of anomalous attacks, this paper also implemented attack methods that were not used earlier. To verify the effectiveness of the model in a real-world application scenario, the authors collected raw network traffic packets for one week.

Regarding the processing of packets: first, the packets are collated and filtered for information such as port and length. Then, the data flow is traced and the packets are parsed. To avoid a significant loss of anomalous features during data pre-processing, the authors use Wireshark directly to obtain binary information about the network and transport layers in the traffic packets. This binary sequence is then processed using One-hot encoding. The binary information is eventually presented in hexadecimal, with each byte taking the shape of [1,257] after processing. This information is then fed into the model as a sequence. Finally, the traffic data is labeled. The dataset contains labels for both normal and abnormal attacks. The number of records in the training set is 30,000 and in the test set 10,000. The processing of the data is shown in Figure 8:

Figure 8. Data processing



```
Frame 4: 189 bytes on wire (1512 bits), 189 bytes captured (1512 bits)
Ethernet II, Src: LCFCHeFe_b6:8e:c2 (54:05:db:b6:8e:c2), Dst: Hangzhou_29:7b:4a (28:57:be:29:7b:4a)
Internet Protocol Version 4, Src: 192.168.1.235, Dst: 192.168.1.65
Transmission Control Protocol, Src Port: 3792, Dst Port: 80, Seq: 1, Ack: 1, Len: 135
Hypertext Transfer Protocol
```

Application

The video surveillance network intrusion detection task is defined as a binary classification task that detects anomalous attacks. Table 9 shows the results of applying the TCN-Attention model to a practical application scenario of video-specific network intrusion detection, where the TCN-Attention model performs best. Faced with anomalous network attacks in a specific application scenario, the model is able to learn the features of different attack methods well and complete the intrusion detection task. The above application can prove that the TCN-attention model has excellent network attack detection capability. It can be widely used in real life to provide services for the public network security of society.

Table 9. Comparison of evaluation metrics for all models

Baseline	Wa-precision	Wa-recall	Wa-F1 score	Accuracy
TCN-Attention	**94.45%**	**90.83%**	**91.72%**	**93.10%**
TCN	92.32%	87.59%	89.46%	91.97%
CNN-LSTM	91.68%	91.47%	87.67%	91.87%
CNN	86.67%	75.03%	77.78%	83.33%
LSTM	66.67%	66.67%	53.33%	66.67%
MLP	64.12%	67.50%	63.23%	63.68%
RNN	66.67%	66.67%	53.33%	66.67%
GRU	66.66%	66.67%	53.33%	66.67%

CONCLUSION

The TCN intrusion detection model incorporating the attention mechanism is proposed in this work. The text data is first converted and normalized into numeric types. Then the One-hot encoded data is fed into the TCN network. After, the convolution and residual modules can achieve long-term memory and extraction of spatial and temporal information of higher-order network traffic features. The attention mechanism can further help the model focus on the feature attributes corresponding to different attacks and better solve the multi-classification problem. Since the TCN model is a variant of a convolutional neural network, it has the advantages of a convolutional model and, in addition, the ability of a temporal model to learn from long-term memory of the data. Therefore, to verify the performance of the model proposed in this paper, the authors chose to test the model using the classical temporal models RNN, LSTM, GRU, the convolutional model CNN, the concise deep learning model MLP and the fusion model CNN-LSTM for comparison. The dataset used for the comparison is the KDD CUP99 10% data subset and UNSW-NB15. These are two classical network intrusion detection datasets and have had duplicates and redundancies removed from the original data to ensure the validity of the model's detection results. The performance advantages of the model are verified by comparing the evaluation metrics of accuracy, weighted average accuracy, recall, and F1 Score. Finally, the model's broad application capability is demonstrated by applying it to anomaly detection in a practical scenario in a video surveillance network.

Our model gives good results in applications, but its detection of anomalous categories with fewer attack records is not yet satisfactory. In the future, the work will first start to solve the data imbalance problem, improve the detection rate of anomalous attack types with fewer records. Secondly, try to validate and enhance the model's detection efficacy on more novel attacks. Finally, solving the overfitting problem that tends to occur in the training of the model. Ultimately, the ability of the model to handle unbalanced data is improved.

ACKNOWLEDGMENT

This research was supported by China Postdoctoral Science Foundation [grant number 2020M670998];Shanghai Talent Development Fund under Grant [grant number 2020016], the Natural Science Foundation of Shanghai [grant number 21ZR1422000]; and the Key R & D program of Sichuan [grant number 2021YFS0310].

REFERENCES

Bai, S., Kolter, S., & Koltun, V. (2019). Trellis Networks for Sequence Modeling. *7th International Conference on Learning Representations(ICLR).*

Benjamin, J., Radford, B., Leonardo, M., & Antonio, J. (2018). Network Traffic Anomaly Detection Using Recurrent Neural Networks. *ArXiv, 1803.10769.*

Chen, H., & Chen, J. (2019). Recurrent Neural Networks Based Wireless Network Intrusion Detection and Classification Model Construction and Optimization. *Dianzi Yu Xinxi Xuebao*, *41*, 1427–1433.

Cho, K., Merrienboer, B., Gulcehre, C., Bahadanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *ArVix*, *1406*, 1724–1734.

Fan, Y., Li, C., Yi, Q., & Li, B. (2021). Classification of Field Moving Targets Based on Improved TCN Network. *Computer Engineering*, *5*, 1–12.

Guirguis, K., Schorn, C., Guntoro, A., Abdulatif, S., & Yang, B. (2021). SELD-TCN: Sound Event Localization & Detection via Temporal Convolutional Networks. *2020 28th European Signal Processing Conference (EUSIPCO), Visual Conference*, 16-20.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. doi:10.1162/neco.1997.9.8.1735 PMID:9377276

Hurley, T., Perdomo, J., & Perez-Pons, A. (December 2016). HMM-Based Intrusion Detection System for Software Defined Networking. *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 617-621.

Kalchbrenner, N., Espeholt, L., Simonyan, K., Oord, A., Graves, A., & Kavukcuoglu, K. (2016). Neural Machine Translation in Linear Time. *ArXiv, 1610.10099.*

Kim, J., Thu, H., & Kim, H. (2016). Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection. *2016 International Conference on Platform Technology and Service (PlatCon)*, 1-5. doi:10.1109/PlatCon.2016.7456805

Lea, C., Vidal, R., Reiter, A., & Hager, G. (2016). Temporal convolutional networks: A unified approach to action segmentation. *14th European Conference on Computer Vision(ECCV).*

Li, J., Xia, S., Lan, H., Li, S., & Sun, J. (2021). Network intrusion detection method based on gated recurrent unit and recurrent neural network. *Journal of Harbin Engineering University,* 1-6.

Lin, S., Shi, Y., & Xue, Z. (2018). Character-Level Intrusion Detection Based On Convolutional Neural Networks. *2018 International Joint Conference on Neural Networks (IJCNN)*, 1-8. doi:10.1109/IJCNN.2018.8488987

Liu, J., Su, P., Yang, M., He, L., Zhang, Y., Zhu, X., & Lin, H. (2018). Software and Cyber Security——A Survey. *Journal of Software*, *29*, 42–68.

Liu, J., Sun, X., & Jin, J. (2020). Intrusion detection model based on Principal Component Analysis and Recurrent Neural Network. *Journal of Chinese Information Processing*, *34*, 105–112.

Liu, J., Yin, L., Hu, Y., Lv, S., & Sun, L. (2018). A Novel Intrusion Detection Algorithm for Industrial Control Systems Based on CNN and Process State Transition. *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*, 1-8.

Liu, Q., Li, Y., Guo, T., & Li, Y. (2021). Intrusion Detection Method Based on Borderline-SMOTE and Double Attention. *Computer Science*, *48*, 327–332.

Liu, T., Cai, S., Yang, H., & Zhang, C. (2019). Network Intrusion Detection Method Integration CNN and BiLSTM. *Computer Engineering*, *45*, 127–133.

Manickam, M., Ramaraj, N., & Chellappan, C. (2017). A Combined PFCM and Recurrent Neural Network based Intrusion Detection System for Cloud Environment. *International Journal of Business Intelligence and Data Minin*, *14*(4), 504–527. doi:10.1504/IJBIDM.2019.099963

Naseer, S., & Saleem, Y. (2018). Enhanced Network Intrusion Detection using Deep Convolutional Neural Networks. *Transactions on Internet and Information Systems (Seoul)*, *12*, 5159–5178.

Ngo, T., Rustia, D., Yang, E., & Lin, T. (2021). Honey Bee Colony Population Daily Loss Rate Forecasting and an Early Warning Method Using Temporal Convolutional Networks. *Sensors (Basel)*, 21.

Ramchoun, H., Idrissi, M., & Ettaouil, M. (2016). Multilayer Perceptron: Architecture Optimization and Training. *International Journal of Artificial Intelligence Tools*, *4*, 26–30.

Reddy, R., Ramadevi, Y., & Sunitha, K. (2016). Effective discriminant function for intrusion detection using SVM. *2016 International Conference on Advances in Computing Communications and Informatics (ICACCI)*, 1148-1153. doi:10.1109/ICACCI.2016.7732199

Roy, B., & Cheung, H. (2018). A Deep Learning Approach for Intrusion Detection in Internet of Things using Bi-Directional Long Short-Term Memory Recurrent Neural Network. *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, 1-6.

Shu, H., Wang, C., & Shi, Y. (2020). Intrusion detection based on BiLSTM and attention mechanism. *Computer Engineering and Design*, *41*, 3042–3046.

Teng, L., Teng, S., Tang, F., Zhu, H., Zhang, W., Liu, D., & Liang, L. (2014). A Collaborative and Adaptive Intrusion Detection Based on SVMs and Decision Trees. *2014 IEEE International Conference on Data Mining Workshop (ICDMW)*, 898-905. doi:10.1109/ICDMW.2014.147

Tian, G., Shan, Z., Liao, Z., & Wang, Y. (2021). Network intrusion detection model based on Faster R-CNN deep learning. *Journal of Nanjing University of Science and Technology*, *45*, 56–62.

Wang, Y., Ma, J., Sharma, A., Singh, P., & Baz, M. (2021). An Exhaustive Research on the Application of Intrusion Detection Technology in Computer Network Security in Sensor Networks. *Sensors (Basel)*, *2021*, 1–11.

Xiao, Y., Xing, C., Zhang, T., & Zhao, Z. (2019). An Intrusion Detection Model Based on Feature Reduction and Convolutional Neural Networks. *IEEE Access: Practical Innovations, Open Solutions*, *7*, 42210–42219.

Yan, B., & Han, G. (2018). LA-GRU: Building Combined Intrusion Detection Model Based on Imbalanced Learning and Gated Recurrent Unit Neural Network. *Security and Communication Networks*, *2018*, 1–13. doi:10.1155/2018/6026878

Yan, J., Mu, L., Wang, L., Ranjan, R., & Zomaya, A. (2020). Temporal Convolutional Networks for the Advance prediction of ENSO. *Scientific Reports*, *10*, 1–15.

Yao, J., Cheng, C., Han, J., & Liu, V. (2021). Anomaly detection method based on multi-task temporal convolutional networks in cloud workflow. *Jisuanji Yingyong*, *5*, 1–9.

Yao, R., Wang, N., Liu, Z., Chen, P., & Sheng, X. (2021). Intrusion Detection System in the Advanced Metering Infrastructure: A Cross-Layer Feature-Fusion CNN-LSTM-Based Approach. *Sensors (Basel)*, *21*, 626.

Yin, C., Zhu, T., & He, X. (2017). A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. *IEEE Access: Practical Innovations, Open Solutions*, *5*, 21954–21961. doi:10.1109/ACCESS.2017.2762418

Yu, F., & Koltun, V. (2016). Multi-Scale Context Aggregation by Dilated Convolutions. *4th International Conference on Learning Representations(ICLR).*

Zaremba, W., Sutskever, I., & Vinyals, O. (2014). Recurrent Neural Network Regularization. *Arxiv*, *1409*, 2329.

Zhai, J. (2021). Research on timing data based on TCN. *Electronic Technology&Software Engineering*, *2*, 196–198.

Zhang, K., & Liu, J. (2021). Attack Path Analysis Method Based on Absorbing Markov Chain. *Computer Science*, *48*, 294–300.

Zhang, Q., & Wang, H. (2021). Intrusion detection model based on dilated convolution and gated recurrent unit. *Jisuanji Yingyong*, *5*, 1–8.

Ping Zhao is currently pursuing the master’s degree in People’s Public Security University of China. Her research interests include Cyber Security, Video networks, and Deep Learning.

Zhijie Fan received a Ph.D. degree from Tongji University, Shanghai, China, in 2019. He joined the faculty of software school, Fudan University in 2019. Now he is a post-doctoral at Fudan University. His current research interests include network security and machine learning.

Zhiwei Cao received a Ph.D. degree from Tongji University, Shanghai, China, in 2021. He is an Assistant Researcher with the Information Security Technology Division, Third Research Institute of Ministry of Public Security, Shanghai. His current research interests include information security, machine learning, and complex networks.

Xin Li received a Ph.D. degree from Zhejiang University, Hangzhou, China, in 2006. He is currently the Professor of Computer Science, People’s Public Security University of China. His research interests include Big Data, Cyber Security, and Video Content Analysis.