IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 31, 2021, accepted July 31, 2021, date of publication August 13, 2021, date of current version September 10, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3104718

Method of Network Intrusion Discovery Based on Convolutional Long-Short Term Memory Network and Implementation in VSS

ZHIJIE FAN[1,2] **AND ZHIWEI CAO**[2]

[1]School of Computer Science, Fudan University, Shanghai 200433, China
[2]Information Security Technology Division, Third Research Institute of Ministry of Public Security, Shanghai 201204, China

Corresponding author: Zhiwei Cao (zhiweicao@126.com)

This work was supported in part by China Postdoctoral Science Foundation under Grant 2020M670998, in part by the Natural Science Foundation of Shanghai under Grant 21ZR1422000, and in part by the Key Research and Development Program of Sichuan under Grant 2021YFS0310.

ABSTRACT Network intrusion discovery aims to detect the network attacks and abnormal network intrusion efficiently, that is an important protection implement in the field of cyber security. However, the traditional network intrusion discovery method are difficult to extract high-order features (such as spatial-temporal information) from network traffic data. In this paper, we proposed an improved method of network intrusion discovery based on convolutional long-short term memory network. This method implements the convolution operation in deep learning into the network structure of long-short term memory and improves the accuracy of network intrusion discovery. In the experimental section, we compared with other similar methods, the result shows that the proposed method has some advantages in the aspects of overall network intrusion discovery index, detection index of different types, and AUC evaluation index. In addition, we applied our method to the network intrusion discovery scenarios of video surveillance system (VSS). The result shows that the proposed method has advantages in accuracy, recall, precision, and other similar methods.

INDEX TERMS Network security, network intrusion discovery method, deep learning, convolutional long-short term memory network.

I. INTRODUCTION

With the rapid development of the internet, a huge number of cyber security events are springing up. It is necessary to propose a new security protection measure to ensure network security [1]–[3].

Network intrusion discovery technology is one of the important measures in the field of Internet security, and will certainly pay more attention under the new situation. It mainly establishes the evaluation and classification model by collecting the data information of key nodes in the network environment, and then judges whether the network access is abnormal. In the field of industrial applications, network intrusion discovery [4] is a powerful supplement to traditional network security protection methods, such as firewalls and anti-viruses, and can fully guarantee the system integrity and data security of key hosts in network data.

In recent years, there are many algorithms on network intrusion discovery, which can be roughly divided into five categories: software defined algorithms [5], [6], decision tree classification algorithms [7], [8], clustering-based algorithms [9], [10], machine learning algorithms [11], [12], neural network algorithms [13], [14], etc. These algorithms have improved the ability of network intrusion discovery in network environment to a certain extent, yet each has some defects [15]. For example, the decision tree classification algorithm has fewer parameters in the model, but it has a serious overfitting problem, which leads that it is difficult to guarantee the accuracy of network intrusion discovery. The machine learning algorithms have better learning efficiency in small samples, but are difficult to extract the high-level features of the data. The intelligent optimization algorithms simulate the relevant habits of various organisms in nature, but the performance ability is poor when the deviation of attack characteristics is large. However, the deep learning can extract the high-dimensional features and discover the

The associate editor coordinating the review of this manuscript and approving it for publication was Yonghong Peng.

relationship between features, with relatively high learning efficiency and network intrusion discovery accuracy.

In recent years, the deep learning has been applied to the field of network intrusion discovery. For example, Kong [16] research on network intrusion discovery algorithm based on network anomaly. Compared with the traditional machine learning methods, this method has greatly reduced the false positive rate and the false negative rate. Shi *et al.* [17] applied recurrent neural network to network intrusion discovery with hessian free optimization, and verified it with KDD Cup 99 data set. Fan *et al.* [18] research on network intrusion discovery of industrial control system based on Long Short-Term Memory (LSTM). Xiao *et al.* [19] proposed a malware detection model based on LSTM network and attention mechanism. Although the above researches are better than the traditional machine learning algorithms and the intelligent optimization algorithms, how to build a neural network structure which include spatial-temporal information is more in line with the problem. Moreover, most studies only extract a single feature from the network traffic data. For example, some papers only consider the space feature [16], and some papers only consider the temporal information [17]. The researchers [18]–[20] has proposed several methods of network intrusion discovery based on spatial-time analysis and applied them to SDN. However, the spatial-temporal information is equally important in network intrusion discovery, and the deep learning method is lacking.

In this paper, we compare some different neural networks. Moreover, it is essential to measure which model is the best choice for network intrusion discovery. The main works of this paper are described as follows:

- ConvLSTM-based network intrusion discovery method was proposed, it considers the spatial and temporal characteristics of the data at the same time and reduces the error caused by the huge difference in the number of samples.
- Experiment was implemented by comparing with other deep learning methods using KDD Cup 99 datasets.
- Application in video surveillance system (VSS) was implemented.

The rest of paper is organized as follows. Section II summarizes the related work. Section III proposes an improved network intrusion discovery method. Section IV introduces the dataset and all evaluation metrics, and shows the results of the experiment. Section V apply the proposed method to VSS. Finally, Section VI concludes the paper.

II. RELATED WORK

Deep neural networks have been successfully applied to various fields (i.e., Computer Vision, Natural Language Processing (NLP), etc.) in recent years. The most relevant works to network intrusion discovery are mainly focused on Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM). In addition, in order to verify the effectiveness of the proposed model, we also compared some other classical neural network models.

Multi-Layer Perceptron (MLP) is a neural network only with the feedforward [21]. The network with the shallow and simple structure, and the activation functions are also simple. Each neuron has a specific activation function, and each connection of two neurons represents a weight for the signal passing through the connection, which is the memory of the MLP model. A typical MLP model includes an input layer, an output layer and a hidden layer or multiple hidden layers. The operation between layers is equal to the fully connected, and the activation function is Sigmoid.

Recurrent Neural Network (RNN). The input data is sequential, which recursive processing is performed in the evolution direction of the sequence and all nodes (cyclic units) are connected in a chain. The current output of a sequence is also related to the previous output. The output value of the previous layer will be added to the input value of the latter layer, which is a connection between the hidden layers. A fully connected RNN satisfies the general approximation theorem, a fully connected recurrent neural network can approximate any nonlinear system, and there is no restriction on the compactness of the state space, if it has enough nonlinear nodes. On this basis, any Turing computable function can be calculated by a finite dimensional full connection, so the RNN is the result of Turing completeness.

Simple Recurrent Network (SRN) [22] is a simple RNN only with a hidden layer. After the back propagation algorithm was proposed [23], the academia began to train the recurrent neural network under the BP framework [24]. The researchers [25], [26] has proposed several methods of network intrusion discovery based on complex network analysis. In 1989, Ronald Williams and David Zipser proposed a real time recurrent learning (RTRL) for RNN [27]. Then, Paul Werbos proposed a BP through time (BPTT) algorithm in 1990 [28]. In 1991, Sepp Hochreiter discovered the long-term dependence problem of recurrent neural networks, which is the recurrent neural networks will appear the gradient disappearance and gradient explosion phenomenon [29].

Gated Recurrent Unit Network (GRU) [30]. The corresponding loop unit contains only two gates: the update gate and the reset gate. It can achieve the equivalent effect of LSTM, and it is easier to train in comparison, which can greatly improve the training efficiency.

CNN-LSTM [31] is a fusion model based on CNN and LSTM. The hierarchical structure is shown in Figure 1.

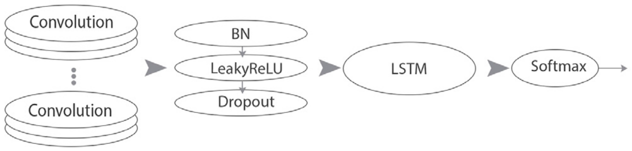


FIGURE 1. Hierarchical structure of CNN-LSTM network.

The feature fusion of CNN-LSTM is shown in Figure 2. It is based on the cross-layer feature fusion of CNN and

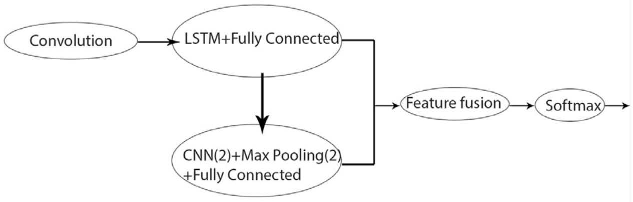


FIGURE 2. Feature fusion of CNN-LSTM network.

LSTM. Since the spatial and temporal characteristics of network traffic data are extracted separately, there are loss and incompatibility of the feature information.

The CNN extracts the high-level features of the input data by convolution operation. In addition, in order to speed up the network learning rate and avoid the problems such as gradient disappearance and gradient explosion, each convolution layer is followed by a Batch Normal layer, and a LeakyReLU activation function. Then the advanced features extracted by the CNN layer are input into the LSTM, and the classification result is obtained through a Softmax function.

Compared with the CNN-LSTM model, the ConvLSTM model introduces the convolution operation into the internal structure of the LSTM, so that the high-order spatial-temporal information of the traffic data can be better combined, and the time sequence relationship between features can be well taken into account on the premise of extracting the spatial features of the data.

In order to solve the problem of long-term dependence, the improvement of RNN appears constantly, including the neural history compressor (NHC) [32], [33] and the long short-term memory networks, which were proposed by Jurgen schmidhub and his collaborators in 1992 and 1997. The LSTM model uses the gating function to determine the forgetting and memory of historical data, and is mainly used to process the data with temporal characteristics. For example, LSTM is used in continuous handwriting recognition, and is also widely used in autonomous speech recognition. Although the LSTM model can connect contextual information well, it requires a lot of calculations for large amounts of the input data, which reduces the efficiency of information interaction between the upper and lower network traffic data, and reduces the accuracy of the algorithm. To remedy the drawback, it is necessary to introduce the CNN model, which can better extract the multi-dimensional features of the data through convolution operation and pooling operation.

III. METHODOLOGY

In this paper, we selected 6 different neural network models for comparison, including Multi-Layer Perceptron (MLP), Recurrent Neural Network (i.e., SRN, LSTM, GRU), CNN-LSTM and CNN.

A. CONVOLUTIONAL NEURAL NETWORK

Deep neural network includes the input layer, the hidden layer and the output layer. The hidden layer [34], [35] have convolutional layer, pooling layer, activation function and batch normalize, etc. The CNN model can better extract the multi-dimensional features of the data by convolution and pooling operations, and has been successfully applied to many fields. The structures of CNN are varies, such as VGG [36], Resnet [37], YOLO [38]–[41], and so on, yet they always include the typical 'CP' structures, which is the Convolutional layer and the Pooling layer during features extraction. The Fully connected layer plays the role of comprehensive features, but it contains a huge of parameters and occupies more memory. In addition, in order to obtain better classification ability, various nonlinear activation functions are proposed, such as sigmoid, ReLU [42], LeakyReLU [43], etc.

Convolutional layer can obtain various features based on different channels and kernels, and then reduce the network parameters by sharing weights [44]. In addition, the input data needs to be preprocessed for network adaptability. The convolutional neural network extracts the local features by the convolution operation, and then synthesizes it, which not only obtains the global features, but also reduces the number of neuron nodes [45]–[48].

Pooling layer is used to remove the redundant information, and improve the convergence speed for the network. The scale invariance is its classic characteristic, which can reduce the network parameters while retaining important information. The types of pooling layer include the Min Pooling, the Average Pooling and the Max Pooling. The important information usually has great value, so the MaxPool [49] exists in numerous deep neural networks, yet the Adaptive MaxPool [50] and the Spatial Pyramid Pooling [51] are becoming fashionable.

Fully connected layer maps the distributed features representation to the sample labeling space. Although it has the function for comprehensive features, it occupies a larger proportion of the parameters in the entire network. Due to the more redundant information and the calculations, it's outlook glummer. Recently, some excellent network models, such as ResNet and GoogLeNet [52]–[54], use the global average pooling (GAP) [55] instead of the Fully Connected to fuse the depth features. Finally, they still use Softmax as the network objective function, which can guide the learning process.

Non-linear activation function makes the high dimensional complex data separable, and prevents the gradient from disappearing. It is an important part of the deep neural network structure. However, a linear activation function will be generally used as a classifier for the last layer in the network.

B. LONG SHORT-TERM MEMORY NETWORK

In 1997, Long Short-Term Memory [56] was first proposed. Due to its unique structure, it is suitable for processing and predicting important events with very long interval and delay in time series. It is not only widely used in natural language processing, but also can be used as a complex nonlinear unit to construct larger deep neural networks.

An LSTM model has three gates to protect and control the cell state. In that unit, like Figure 3, the forget gate and the

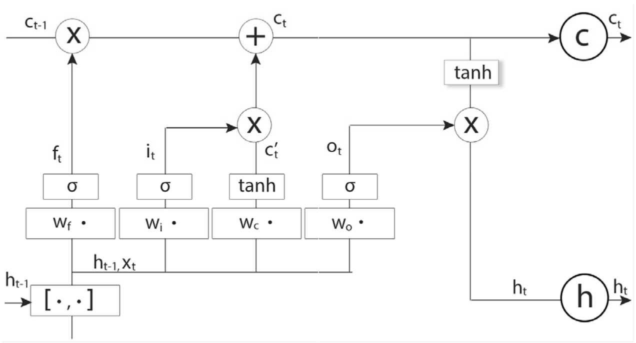


FIGURE 3. Cell structure of LSTM network.

input gate play a significant role at the same time. Here f_t and i_t denote the forget gate and the input gate, respectively. C' represents the "new candidate values" and C represents the "cell state".

1) FORGET GATE LAYER

The layer decides what information to discard from the cell state, and this decision is made by a sigmoid layer. In Figure 4, it inputs h_{t-1} and x_t, and outputs a number between 0 and 1 in the cell state C_{t-1}. 1 represents the "completely keep this", while 0 represents the "completely get rid of this".

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right), \tag{1}$$

where x_t is the input of data in the current state, h_{t-1} represents the input received from the previous node, W_f and b_f are the weight and the bias, respectively.

2) INPUT GATE LAYER

The layer decides what new information to store in the cell state, and it has two parts. First, a sigmoid layer called the "input gate layer" decides which values we'll update. Next, a *tanh* layer creates a vector of the new candidate values, C', that could be added to the state.

$$\mathrm{i_t} = \sigma\left(\mathrm{W_i} \cdot [\mathrm{h_{t-1}}, \mathrm{x_t}] + \mathrm{b_i}\right), \tag{2}$$

$$\mathrm{C'} = \tanh\left(\mathrm{W_c} \cdot [\mathrm{h_{t-1}}, \mathrm{x_t}] + \mathrm{b_c}\right), \tag{3}$$

3) UPDATE OLD CELL

Now, it is necessary to update the old cell state, C_{t-1}, into the new cell state, C_t.

The previous steps have already determined what to do. We need multiply the old state by f_t, which can help to forget the things that we decide to forget. Then we add the new candidate values, $(i_t \circ C')$, which is scaled by how much that we decide to update.

$$\mathrm{C_t} = \mathrm{f_t} \circ \mathrm{C_{t-1}} + \mathrm{i_t} \circ \mathrm{C'}, \tag{4}$$

4) OUTPUT LAYER

The output layer will be based on the cell state. First, we need to run a sigmoid layer, which decides what parts of the cell state to output. Then, we put the cell state through a *tanh* function which pushing the value to be between −1 and 1, and then multiply it by the output of sigmoid gate.

$$\mathrm{O_t} = \sigma\left(\mathrm{W_0} \cdot [\mathrm{h_{t-1}}, \mathrm{x_t}] + \mathrm{b_o}\right), \tag{5}$$

$$\mathrm{h_t} = \mathrm{O_t} \circ \tanh\left(\mathrm{C_t}\right), \tag{6}$$

In 1999, Gers & Schmidhuber [57] proposed a popular LSTM variant, which added a "peephole connections" on the basis of the traditional LSTM model. In Figure 4, we find that the process can retain the previous cell state.

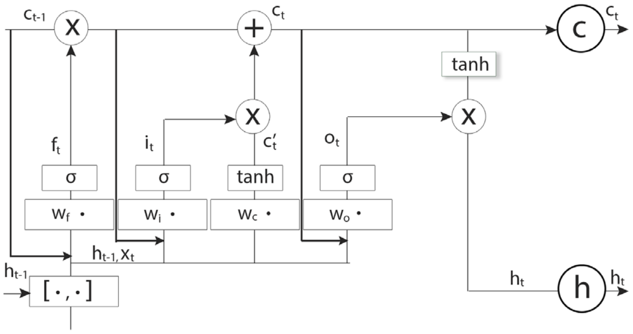


FIGURE 4. Cell structure of LSTM with peephole connections.

The above diagram adds peepholes to all the gates, and the process is changed as follows:

$$\mathrm{f_t} = \sigma\left(\mathrm{W_f} \cdot [\mathrm{h_{t-1}}, \mathrm{x_t}] + \mathrm{W_f} \circ \mathrm{C_{t-1}} + \mathrm{b_f}\right), \tag{7}$$

$$\mathrm{i_t} = \sigma\left(\mathrm{W_i} \cdot [\mathrm{h_{t-1}}, \mathrm{x_t}] + \mathrm{W_i} \circ \mathrm{C_{t-1}} + \mathrm{b_i}\right), \tag{8}$$

$$\mathrm{O_t} = \tanh\left(\mathrm{W_o} \cdot [\mathrm{h_{t-1}}, \mathrm{x_t}] + \mathrm{W_o} \circ \mathrm{C_{t-1}} + \mathrm{b_o}\right), \tag{9}$$

where ∘ is Hadamard product, · is Matrix multiplication, other terms are consistent with the above.

C. ConvLSTM NETWORK

ConvLSTM is a new network structure based on the LSTM and the Convolution operation. The cell unit also include the Input gate, the Forget gate and the Output gate [17], [58]. Compared with CNN-LSTM, it replaces the matrix multiplication with the convolution operation. The method can obtain spatial-temporal characteristics at the same time, and retain the relevant information of both. Moreover, due to the convolution operation with sharing weights, it can reduce the parameters and time complexity. The process is as follows:

$$\mathrm{i_t} = \sigma\left(\mathrm{W_{xi}} * \mathrm{x_t} + \mathrm{W_{hi}} * \mathrm{h_{t-1}} + \mathrm{W_{ci}} \circ \mathrm{C_{t-1}} + \mathrm{b_i}\right), \tag{10}$$

$$\mathrm{f_t} = \sigma\left(\mathrm{W_{xf}} * \mathrm{x_t} + \mathrm{W_{hf}} * \mathrm{h_{t-1}} + \mathrm{W_{cf}} \circ \mathrm{C_{t-1}} + \mathrm{b_f}\right), \tag{11}$$

$$\mathrm{C_t} = \mathrm{f_t} \circ \mathrm{C_{t-1}} + \mathrm{i_t} \circ \tanh\left(\mathrm{W_{xc}} * \mathrm{x_t} + \mathrm{W_{hc}} * \mathrm{h_{t-1}} + \mathrm{b_c}\right) \tag{12}$$

$$\mathrm{O_t} = \sigma\left(\mathrm{W_{xo}} * \mathrm{x_t} + \mathrm{W_{ho}} * \mathrm{h_{t-1}} + \mathrm{W_{co}} \circ \mathrm{C_t} + \mathrm{b_o}\right), \tag{13}$$

$$\mathrm{h_t} = \mathrm{O_t} \circ \tanh\left(\mathrm{C_t}\right), \tag{14}$$

where ∗ is the Convolution operation, other terms are consistent with the above.

D. IMPLEMENTATION

The LSTM and the CNN model are good at processing contextual and space information, respectively. However, the LSTM network structure is complicated, and its computational complexity will increase sharply when the input data is massive. In addition, the enormous number of parameters can be reduced by convolution operation. In this paper, we will propose an improved network intrusion discovery method, which must consider the spatial and temporal information of the network intrusion discovery data set. In term of this, we think the ConvLSTM model is a suitable choice. At the same time, the convolution operation introduced in the ConvLSTM model can accelerate the convergence speed of the entire model, which is suitable for large-scale intrusion data. The process are as follows:

1) TRAFFIC DATA ACQUISITION

The real-time network traffic data is obtained by using the network traffic collection module, and then the characteristics of the traffic data are analyzed, such as the types of service and protocol, the network connection time and connection status of the extracted data.

2) TRAFFIC DATA PREPROCESSING

For discrete feature in the data set, such as the connection status and the service type in the network traffic characteristics, 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, such as the network connection time in the characteristics of the traffic data, normalization is performed in order to ensure that all values in every numeric column are between 0 and 1. The formula of normalization is as follows:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{15}$$

where $\mathrm{x_{min}}$ and $\mathrm{x_{max}}$ are the minimum and the maximum values of the feature vector.

3) SEQUENCE FEATURE EXTRACTION

The feature vector of data packet after the preprocessing is fed into the proposed ConvLSTM model, and the spatio-temporal features are obtained through the ConvLSTM model.

4) NETWORK DATA CLASSIFICATION

In the proposed model, we input the above features into the fully connected layer, which can integrate the complete feature information of the data, and finally obtain the classification result of the network intrusion discovery data by the Softmax function.

The proposed network intrusion discovery method is displayed in Figure 5. It contains an input layer, two ConvLSTM layers with the batch normalization and the Dropout layer, a Convolutional3D layer and two Fully connected layers. The dropout rates are set to 0.3 and 0.5, respectively. The activation function is ReLU, and the last classification layer is Softmax.

The proposed model is an end-to-end model, which transform from the input node to the state node. Meanwhile, the spatial characteristics of the data are extracted by the convolution operation. In addition, the input and output of current data are determined by the input gate and the forget gate. Finally, the output of the current node is determined by the output gate. The convolution operation has the advantages of local linkage and weight sharing, which can reduce the time complexity of the LSTM network and accelerate model convergence.

The ConvLSTM model introduces the convolution operation into the LSTM, which can directly input the eigenvalues obtained by the convolution operation into the prediction network structure, and speed up the convergence speed. The parameters of each layer are shown in Table 1.

IV. EXPERIMENTS

The experiment environments include Tensorflow (1.11) and Keras (2.2.4), and can be run on various platforms, such as CPU and GPU. Moreover, we introduce Adadelta optimizer and loss functions based on categorical cross entropy in training, and the epoch is set to 12, the batch size is set to 128. The details are shown in the following table.

A. DATASETS

The KDD Cup 99 data set was collected in the local area network of air force base, MIT Lincoln laboratory in 1998 by the DAPPA ID Evaluation Group [11], [12]. The data set is mainly used for the Knowledge Discovery and Data Mining, yet here used for the network intrusion discovery. The KDD Cup 99 data set can be divided into five data types, including 'Normal', 'DOS', 'R2l', 'U2r', and 'probe'. The specific classifications are shown in Table 3.

In this paper, we choose the "corrected" and "kddcup.data_10_percent_corrected" as the train and test set, respectively. The details are shown in the Table 4.

B. EVALUATION METRICS

Network intrusion discovery can be considered as a multi-class problem, and the confusion matrix is one of the methods to address the classification issue intuitively. Precision, Recall and F1-score are the three classical evaluation metrics [1]–[3]. Their formulas are as follows:

$$Precision_i = \frac{TP_i}{TP_i + FP_i}, \tag{16}$$

$$Recall_i = \frac{TP_i}{TP_i + FN_i}, \tag{17}$$

$$\mathrm{F1{-}Score_i} = \frac{2 * \mathrm{Precision_i} * \mathrm{Recall_i}}{\mathrm{Precision_i} + \mathrm{Recall_i}}, \tag{18}$$

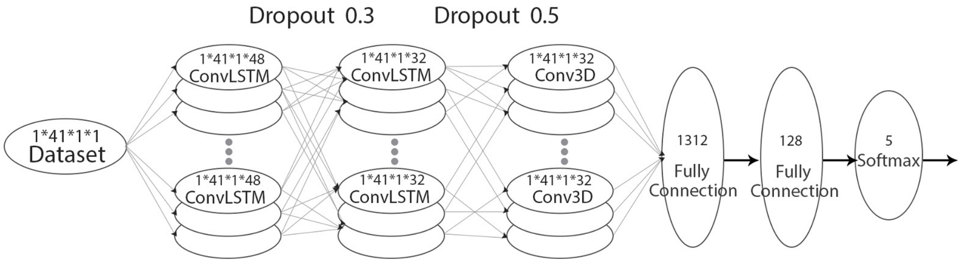


FIGURE 5. Network structure of ConvLSTM.

TABLE 1. The parameters of ConvLSTM model.

Layers	Kernel size	Strides	Rate	Activation	Output size
Input		1			1*41*1*1
Convlstm2d	3*3	1		Relu	1*41*1*48
Dropout			0.3		1*41*1*48
Convlstm2d	3*3	1		Relu	1*41*1*32
Dropout			0.5		1*41*1*32
Conv3d	3*3*3			Sigmoid	1*41*1*32
Dense				Relu	128
Dense				Softmax	5

TABLE 2. Environment and hyper-parameter.

Project	Environment/Hyper-Parameter
Operating System	Ubuntu 18.04
CPU	I5-7200
Memory	8G
GPU	Nvidia 2080Ti
Platform	Tensorflow (1.11) &Keras (2.2.4)
Batch size	128
Epoch	12
Loss function	Cross entropy
Optimizer	Adadelta

where TP is True Positive, which represents the number of correctly identified abnormal samples; FP is False Positive, which represents the number of incorrectly identified abnormal samples; TN is True Negative, which represents the number of correctly identified normal samples; FN is False Negative, which represents the number of incorrectly identified normal samples.

The precision is to calculate how many of the predicted samples are correctly predicted. The recall rate is to calculate how many samples in the entire sample are correctly predicted. It is expected that both Precision and Recall will maintain a relatively high level, but in fact, there are contradictions between the two in some cases. Therefore, F1-Score is a good choice, which consider the precision and recall of the classification model.

In order to reduce the influence caused by the huge difference in the number of samples, we also introduce "Weighted Average" to evaluation metrics, and is denoted as 'WA'. The WA needs to calculate the precision, recall and F1 score of each class in N categories, and then defines the corresponding weights according to the sample number of each class, and finally determines the overall weighted average. Here, α_i represents the weight of class i. The formulas are as follows:

TABLE 3. The specific classifications of KDD cup 99.

Types	Descriptions	Specific classifications
Normal	Normal information	Normal
DOS	Denial of service attack	Neptune, Teardrop, Smurf, Back, Land
Probing	Surveillance and probing	Portsweep, Ipsweep, Nmap, Satan
R2l	Remote to Local attack	Guess_passwd, Warezmaster, Warezclient, Ftp_weite, Multihop, Imap, Phf , Spy
U2r	User to root attack	Buffer_overflow, Loadmodule, Rootkit, Perl

$$WA\text{-}P = \sum_{i=1}^{n} \alpha_i * Precision_i, \tag{19}$$

$$WA\text{-}R = \sum_{i=1}^{n} \alpha_i * Recall_i \tag{20}$$

$$WA\text{-}F1 = \sum_{i=1}^{n} \alpha_i * F1 - Score_i, \tag{21}$$

In this paper, we also select the ROC curve as our metric to evaluate the network intrusion discovery method comprehensively. The ROC curve and AUC are usually used to evaluate the classification problem of imbalanced sample distribution, and FPR and TPR is False Positive Rate and True Positive Rate, respectively. AUC is Area Under ROC Curve.

C. ANALYSIS AND COMPARISONS

Figure 6 shows the prediction of all models in each category. We can see that the proposed ConvLSTM model performed well in comparison to the other models. It is obvious that the ConvLSTM model performs better in 8 out of 15 indicators in five categories. Considering the F1 evaluation metrics, which can better reflect the overall prediction effect of the

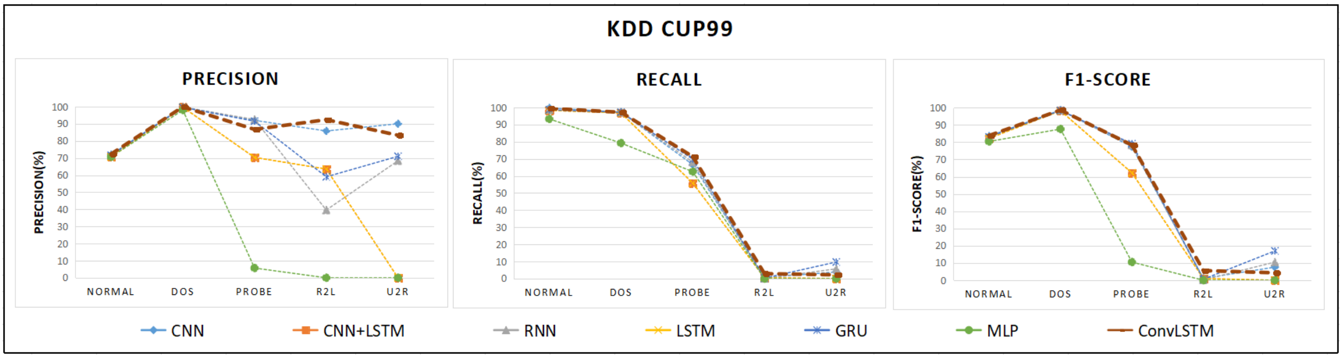


FIGURE 6. Prediction accuracy of all models in each category.

TABLE 4. Details of the train and test sets.

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

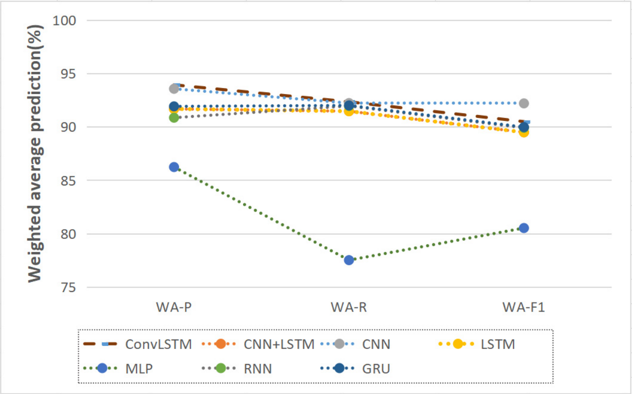


FIGURE 7. Weighted average prediction accuracy of all models in each category.

algorithms, the ConvLSTM model is superior to other classic models in ‘normal’, ‘dos’ and ‘R21’.

The overall weighted average prediction of all compared models is shown in Figure 7. It can be seen that the ConvLSTM model ranks first in the WA-P and WA-R evaluation metrics, and ranks second in the WA-F1 evaluation metric.

The ConvLSTM model can simultaneously obtain the spatial and temporal features of network traffic data, so the precision and the recall rates perform well. In addition, we find that the ability of network intrusion discovery of CNN follows closely behind. This result shows that the features of network traffic data can be well captured by the convolution and pooling operations, but the temporal information is ignored. Therefore, the overall prediction performance is still insufficient. The CNN-LSTM model is a simple connection between CNN and LSTM, which has the feature information loss and incompatible, so the prediction effect of CNN-LSTM model is weaker than the ConvLSTM model. The LSTM, RNN, and GRU models only consider the temporal information of network traffic data, so the overall prediction effects of these models are poor, and need to be improved. The MLP model only has the simple forward structure, which makes it difficult to extract the high-order features of data. Therefore, the prediction of this model is worst.

TABLE 5. The confusion matrix of ConvLSTM.

	Normal	dos	Probe	R2l	U2r
Normal	60347	69	158	18	1
dos	6154	223592	107	0	0
Probe	1193	192	2781	0	0
R2l	15875	1	15	298	0
U2r	141	65	7	0	15

Table 5 shows the confusion matrix of ConvLSTM. We find that the proposed model performs well in ‘Normal’, ‘Probe’, and ‘Dos’. The precision of ‘Normal’ can arrived at 99.5%. However, it performs poorly in ‘R21’ and ‘U2R’. We find the reason by analyzing Table 4. In train set, the sample sizes of ‘Normal’, ‘Dos’ and ‘Probe’ are large, which is enough to obtain the valuable information during training, so the final detection ability is higher relatively. However, the sample sizes of ‘R21’ and ‘U2R’ is too rarer in train set. It is difficult for the model to accurately predict the results of a large sample test set from a small sample train set, resulting in low detection effect in the two categories of ‘R21’ and ‘U2R’. To our knowledge, the confusion matrix can reflect the recall rate, and we find that the proposed model is competitive in the recall rates of five categories.

Figure 8 shows the ROC cures of all models, and the AUC value of each model is also shown. We find that the AUC value of ConvLSTM model is 0.76, which ranks first with CNN, GRU, and RNN models. The result also confirms that the proposed model has high detection and classification ability.

In summary, in order to verify the effectiveness of our method, we select a large number of evaluation metrics, including Precision, Recall, F1-Score, Weighted Average,

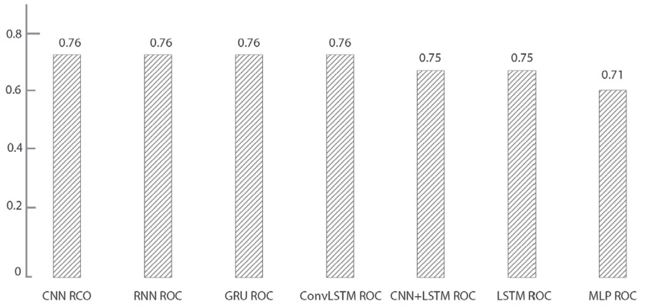


FIGURE 8. **ROC curve results for all models.**

etc. The above-discussed results confirm that the performance of the proposed ConvLSTM model is competitive to the state-of-the-art network intrusion discovery model in various types of evaluation metrics. Therefore, we believe that the ConvLSTM model can extract the high-order spatial and temporal characteristics of network traffic data, so it has better precision in network intrusion discovery.

V. APPLICATION IN VSS

A. INTRUDUCTION

Video surveillance system is used to transmit video images, connect various levels of surveillance systems, and support video-related services. The video surveillance system mainly contains front-end equipment, video processing systems, and various application servers.

Currently, the video surveillance system faces the risks of physical devices, system applications, data information, and network attacks. First, the front-end network devices. Monitoring devices are small, have limited storage. It cannot carry anti-virus software and other protection means. Therefore, the networked devices themselves are prone to vulnerabilities such as weak passwords, override access, SQL injection, buffer overflow. Second, a large number of application system modules in the video surveillance system. Imperfect network supervision, insufficient identity verification, etc., easily cause information leakage and trigger network attacks. Third, the shortcomings of the traditional protection system for video surveillance networks. Firewalls, anti-virus software, fortress machines can hardly cope with the ever-changing means of network attack.

As active defense tools, network intrusion discovery models can monitor network traffic in real-time, sense hidden attacks and analyze various types of attack behaviors. As a result, these tools help maintain network information security and propose corresponding protection strategies.

B. VSS DATASETS

In this work, the original traffic packets were collected from a VSS attacks and defense site in an area of Beijing. It contains a week's worth of traffic data. The types of attacks contained in the data include weak passwords, command injection, device and system vulnerabilities, remote code execution, etc. We studied the front-end equipment, VSS architecture and the application scenarios. All the types of attacks are the most frequent in this scenario and have a significant negative impact on the network. Moreover, the implementation of these attacks is relatively simple. The labels contained in the data are divided into two categories: normal and abnormal. The number of records in the training set is 30000 records and the testing set is 10000 records. Using the Wireshark to obtain network layer and transport layer information in traffic packets. Data is pre-processed by One-hot encoding. The dataset contains labels for both normal and abnormal network intrusion.

C. IMPLEMENTATION

The network intrusion discovery in VSS is defined as a binary classification task that detects anomalous network intrusion. The results of applying our method to a practical application scenario of video-specific network intrusion discovery, where our method performs best. Faced with anomalous network intrusion in a specific application scenario, the method is able to learn the features of different network intrusion methods well and complete the network intrusion discovery task. The above application can prove that our method has excellent network intrusion discovery capability. It can be widely used in real life to provide services for the network security of society.

VI. CONCLUSION

In this paper, we proposed an improved method for the network intrusion discovery. In this method, the network intrusion discovery data is pre-processed by One-Hot coding and normalization, and then the convolution operation is introduced into the internal structure of LSTM, so that the high-order spatial-temporal information of network traffic data can be better fused.

In order to verify the effectiveness of the proposed model, we select six classic neural network intrusion detection models for comparative analysis, such as MLP, CNN, LSTM, RNN, etc. Moreover, we chose the classic network detection dataset, KDD Cup 99. Furthermore, we have carried out a lot of comparative experiments from different aspects, including the overall effect of network intrusion discovery, the detection effect for different types of samples, the confusion matrix results for different types of samples, and the results of ROC curve. In addition, we applied our method to VSS, the results show that the proposed model effectively improves the network intrusion discovery ability of network traffic, and provides a new idea for network intrusion discovery of massive network traffic data.

AUTHOR CONTRIBUTIONS

Conceptualization: Z.F. and Z.C.; methodology: Z.F. and Z.C.; formal analysis: Z.F.; resources: Z.C.; writing original draft preparation: Z.F.; visualization: Z.C.; project administration: Z.C. All authors have read and agreed to the published version of the manuscript.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers and the editors for their insightful comments and assistance.

INSTITUTIONAL REVIEW BOARD STATEMENT

Not applicable.

INFORMED CONSENT STATEMENT

Not applicable.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: [http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html].

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] S. Leyi, Z. Hongqiang, L. Yihao, and L. Jia, "Intrusion detection of industrial control system based on correlation information entropy and CNN-BiLSTM," *J. Comput. Res. Develop.*, vol. 56, no. 11, pp. 2330–2338, 2019.

[2] Y. Ding and Y. Zhai, "Intrusion detection system for NSL-KDD dataset using convolutional neural networks," in *Proc. 2nd Int. Conf. Comput. Sci. Artif. Intell. (CSAI)*, 2018, pp. 81–85.

[3] C. Liu, Y. Liu, Y. Yan, and J. Wang, "An intrusion detection model with hierarchical attention mechanism," *IEEE Access*, vol. 8, pp. 67542–67554, 2020.

[4] D. E. Denning, "An intrusion-detection model," *IEEE Trans. Softw. Eng.*, vol. SE-13, no. 2, pp. 222–232, Feb. 1987.

[5] M. Akbanov, V. G. Vassilakis, and M. D. Logothetis, "Ransomware detection and mitigation using software-defined networking: The case of WannaCry," *Comput. Electr. Eng.*, vol. 76, pp. 111–121, Jun. 2019.

[6] D. Kreutz, F. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[7] M. K. Nanda and M. R. Patra, "Intrusion detection and classification using decision tree-based feature selection classifiers," in *Intelligent and Cloud Computing*. Singapore: Springer, 2021, pp. 157–170.

[8] S. M. Badr, "Adaptive layered approach using C5.0 decision tree for intrusion detection systems (ALIDS)," *Int. J. Comput. Appl.*, vol. 66, no. 22, pp. 18–22, 2013.

[9] Z. Liu, W. Wei, H. Wang, Y. Zhang, Q. Zhang, and S. Li, "Intrusion detection based on parallel intelligent optimization feature extraction and distributed fuzzy clustering in WSNs," *IEEE Access*, vol. 6, pp. 72201–72211, 2018.

[10] A. Appasha Chormale and A. P. Ghatule, "Cloud intrusion detection system using fuzzy clustering and artificial neural network," *J. Phys., Conf. Ser.*, vol. 1478, Apr. 2020, Art. no. 012030.

[11] I. F. Kilincer, F. Ertam, and A. Sengur, "Machine learning methods for cyber security intrusion detection: Datasets and comparative study," *Comput. Netw.*, vol. 188, Apr. 2021, Art. no. 107840.

[12] V. Pai and N. D. Adesh, "Comparative analysis of machine learning algorithms for intrusion detection," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 1013, no. 1, 2021, Art. no. 012038.

[13] S. Ho, S. A. Jufout, K. Dajani, and M. Mozumdar, "A novel intrusion detection model for detecting known and innovative cyberattacks using convolutional neural network," *IEEE Open J. Comput. Soc.*, vol. 2, pp. 14–25, 2021.

[14] G. Andresini, A. Appice, and D. Malerba, "Nearest cluster-based intrusion detection through convolutional neural networks," *Knowl.-Based Syst.*, vol. 216, Mar. 2021, Art. no. 106798.

[15] H. Liu, S. Z. Peng, and B. F. Luo, "Research on intrusion detection model based on ecosystem neural network," *Command Control Simul.*, vol. 42, no. 4, pp. 45–50, 2020.

[16] L. Z. Kong, *Research on Intrusion Detection Algorithm Based on Network Anomaly*. Beijing, China: Beijing Jiao Tong Univ., 2017.

[17] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," 2015, *arXiv:1506.04214*. [Online]. Available: http://arxiv.org/abs/1506.04214

[18] Z. Fan, Y. Xiao, A. Nayak, and C. Tan, "An improved network security situation assessment approach in software defined networks," *Peer Peer Netw. Appl.*, vol. 12, no. 2, pp. 295–309, Mar. 2019.

[19] Y. Xiao, Z.-J. Fan, A. Nayak, and C.-X. Tan, "Discovery method for distributed denial-of-service attack behavior in SDNs using a feature-pattern graph model," *Frontiers Inf. Technol. Electron. Eng.*, vol. 20, no. 9, pp. 1195–1208, Sep. 2019.

[20] Z. Fan, Z. Tan, C. Tan, and X. Li, "An improved integrated prediction method of cyber security situation based on spatial-time analysis," *J. Internet Technol.*, vol. 19, no. 6, pp. 1789–1800, 2018.

[21] E. E. Osuna, *Support Vector Machines: Training and Applications*. Cambridge, MA, USA: Massachusetts Institute of Technology, 1998.

[22] J. L. Elman, "Distributed representations, simple recurrent networks, and grammatical structure," *Mach. Learn.*, vol. 7, nos. 2–3, pp. 195–225, 1991.

[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.

[24] M. I. Jordan, "Serial order: A parallel distributed processing approach," *Adv. Psychol.*, vol. 121, pp. 471–495, May 1997.

[25] Z. Cao, Y. Zhang, J. Guan, S. Zhous, and G. Wen, "A chaotic ant colony optimized link prediction algorithm," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 9, pp. 5274–5288, Sep. 2021, doi: 10.1109/TSMC.2019.2947516.

[26] Z. Cao, Y. Zhang, J. Guan, S. Zhou, and G. Chen, "Link weight prediction using weight perturbation and latent factor," *IEEE Trans. Cybern.*, early access, Jun. 11, 2020, doi: 10.1109/TCYB.2020.2995595.

[27] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, no. 2, pp. 270–280, Jun. 1989.

[28] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.

[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, 1997.

[30] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*. [Online]. Available: http://arxiv.org/abs/1412.3555

[31] R. Lowe, N. Pow, I. Serban, and J. Pineau, "The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems," 2015, *arXiv:1506.08909*. [Online]. Available: http://arxiv.org/abs/1506.08909

[32] R. Yao, N. Wang, Z. Liu, P. Chen, and X. Sheng, "Intrusion detection system in the advanced metering infrastructure: A cross-layer feature-fusion CNN-LSTM-based approach," *Sensors*, vol. 21, no. 2, p. 626, Jan. 2021.

[33] J. Schmidhuber, "Learning complex, extended sequences using the principle of history compression," *Neural Comput.*, vol. 4, no. 2, pp. 234–242, Mar. 1992.

[34] S. Zhang, Y. Gong, and J. Wang, "The development of deep convolution neural network and its applications on computer vision," *Chin. J. Comput.*, vol. 40, no. 9, pp. 1–29, 2017.

[35] Z. K. Wei, M. Cheng, X. B. Zhou, Z. F. Li, B. W. Zou, Y. Hong, and J. M. Yao, "Convolutional interactive attention mechanism for aspect Extraction," *J. Comput. Res. Develop.*, vol. 57, no. 11, p. 2456, 2020.

[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[37] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[38] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[39] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.

[40] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[41] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*. [Online]. Available: http://arxiv.org/abs/2004.10934

[42] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," *Proc. 14th Int. Conf. Artif. Intell. Statist., JMLR Workshop Conf.*, Jun. 2011, pp. 315–323.

[43] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," 2015, *arXiv:1505.00853*. [Online]. Available: http://arxiv.org/abs/1505.00853

[44] T. Sercu, C. Puhrsch, B. Kingsbury, and Y. LeCun, "Very deep multilingual convolutional neural networks for LVCSR," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 4955–4959.

[45] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*. [Online]. Available: http://arxiv.org/abs/1511.07122

[46] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: http://arxiv.org/abs/1704.04861

[47] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[48] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.

[49] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1097–1105.

[50] T. Weiss, J. Hillenbrand, A. Krohn, and F. K. Jondral, "Mutual interference in OFDM-based spectrum pooling systems," in *Proc. IEEE 59th Veh. Technol. Conf. (VTC-Spring)*, vol. 4, May 2004, pp. 1873–1877.

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[52] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[53] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

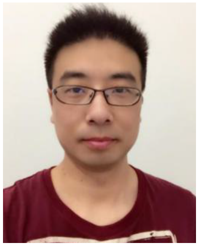
[54] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[55] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*. [Online]. Available: http://arxiv.org/abs/1312.4400

[56] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[57] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, 2000.

[58] Y. Liu, J. J. Cao, and X. C. Diao, "Survey on stability of feature selection," *J. Softw.*, vol. 29, pp. 2559–2579, Jun. 2018.



ZHIJIE FAN received the Ph.D. degree from Tongji University, Shanghai, China, in 2019. He joined the School of Computer Science, Fudan University, in 2019. He is currently holding a postdoctoral position with Fudan University. He is also a Researcher with the Information Security Technology Division, Third Research Institute of Ministry of Public Security, Shanghai. His current research interests include network security and machine learning.

ZHIWEI CAO received the Ph.D. degree from Tongji University, Shanghai, China, in 2021. He is currently 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.