IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

An Improved Method of Network Intrusion Discovery Based on Convolutional Long-short Term Memory Network

Zhijie Fan[1], Zhiwei Cao[2]

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

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

ABSTRACT Intrusion detection model aims to detect abnormal attacks in the network efficiently, which is an important measure in network security protection. Considering that the traditional intrusion detection models are difficult to extract high-order features from network traffic data, an intrusion detection model based on convolutional long-short term memory network is proposed. This model introduces the convolution operation in deep learning into the network structure of long-short term memory, so it can effectively extract the spatial and temporal characteristics of network traffic data, reduce the computational complexity, and improve the accuracy of intrusion detection. First, One-Hot coding and normalization processing are carried out for complex and diverse data features. Then, the higher-order spatial-temporal information in data features is ex-tracted based on convolutional long-short term memory network. Finally, the optimal model is determined by the cross-validation and the principle of single variable, and then achieve the final detection. Compared with other state-of-the-art intrusion detection models, it is concluded that the model has some advantages in the aspects of overall intrusion detection index, detection index of different types, and AUC evaluation index.

INDEX TERMS network security, intrusion detection model, deep learning, convolutional long-short term memory network, One-Hot coding, normalization

I. INTRODUCTION

Internet security is a fundamental and challenging task under the rapid development of 5G, Internet of things, mobile Internet and other communication technologies. Therefore, it is necessary to propose a new security protection measure to ensure network security [1-3].

Intrusion detection 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, intrusion detection [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 intrusion detection, 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 intrusion detection 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 intrusion detection. 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 relationship between features, with relatively high learning efficiency and intrusion detection accuracy.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3104718, IEEE Access

In recent years, the deep learning has been applied to the field of intrusion detection. For example, Kong [16] research on intrusion detection 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. Kim et al. [17] applied recurrent neural network to intrusion detection with hessian free optimization, and verified it with KDD Cup 99 data set. Yu et al. [18] research on intrusion detection of industrial control system based on long short-term memory. Raff 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 [17-19]. However, the spatial-temporal information is equally important in intrusion detection.

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

- We propose an intrusion detection model based on ConvLSTM [20], which considers the spatial and temporal characteristics of the data at the same time.
- We add “Weighted Average” and “ROC curve” to evaluation metrics, in order to reduce the error caused by the huge difference in the number of samples.
- We evaluate the model in KDD Cup 99, and compare them with other deep learning models.

The rest of paper is organized as follows. Section II summarizes the related work. Section III proposes an intrusion detection model. Section IV introduces the dataset and all evaluation metrics, and shows the results of the experiment. Finally, Section V concludes the paper.

II. RELATED WORK

Deep neural networks have been successfully applied to various fields (i.e., Computer Vision, NLP, etc.) in recent years. The most relevant works to intrusion detection are mainly focused on CNN and 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. The structure is shown in Figure 1.

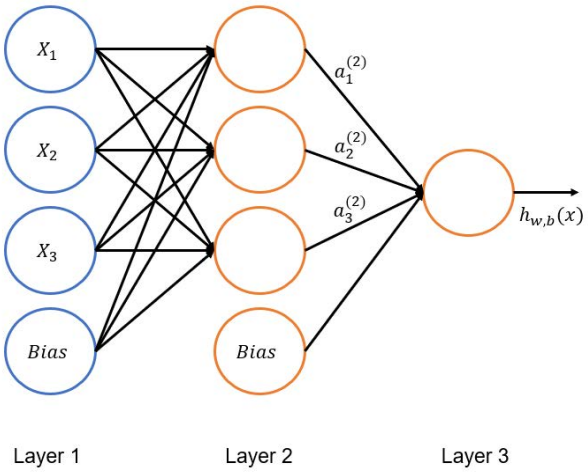

FIGURE 1. The structure of MLP.

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 [24-26]. 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, that is when learning long-term sequences, the recurrent neural networks will appear the gradient disappearance and the gradient explosion phenomenon, and unable to grasp the long-term nonlinear relationship [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. The function of the reset gate is same to the input gate of the LSTM unit, while

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3104718, IEEE Access

the update gate realizes the functions of the forget gate and the output gate at the same time.

CNN-LSTM [31] is a fusion model based on CNN and LSTM. First, the spatial high-order features of the traffic data are extracted, and then re-input into the LSTM model, and there are loss and incompatibility of the feature information before and after. The hierarchical structure is shown in Figure 2.

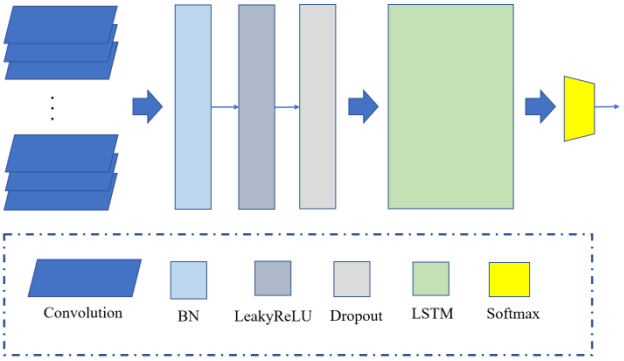

FIGURE 2. Hierarchical Structure of CNN-LSTM Network.

The feature fusion of CNN-LSTM is shown in Figure 3. It is based on the cross-layer feature fusion of the convolutional neural networks (CNN) and the long short-term memory (LSTM) networks. However, it still has spatial-temporal information incoherence.

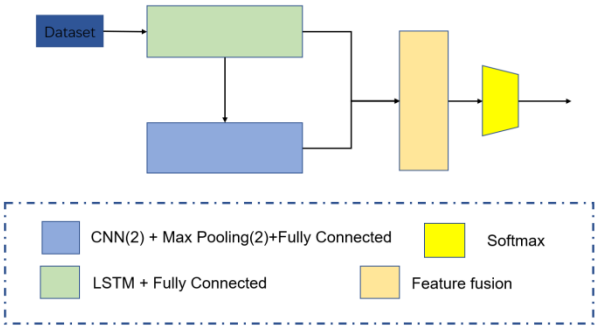

FIGURE 3. Feature Fusion of CNN-LSTM Network.

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. METHOD

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3104718, IEEE Access

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.

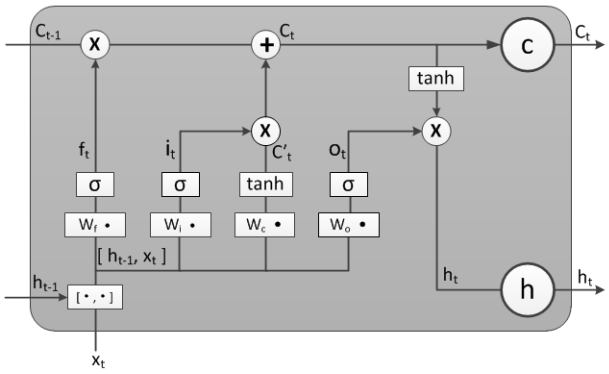

FIGURE 4. Cell structure of LSTM network.

An LSTM model has three gates to protect and control the cell state. In that unit, like Figure 4, the forget gate and the 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 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.

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

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

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.

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

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

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 5, we find that the process can retain the previous cell state.

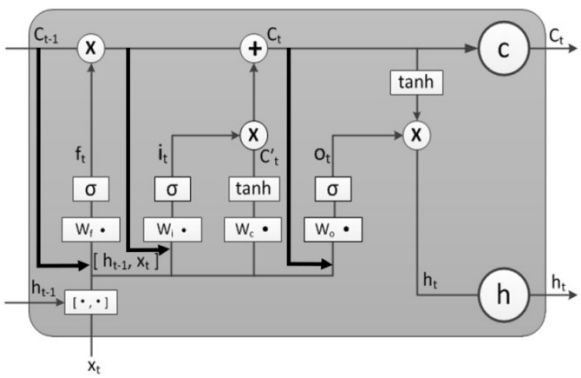

FIGURE 5. Cell structure of LSTM with peephole connections.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3104718, IEEE Access

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

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

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

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

Where $\circ$is Hadamard product, and $\cdot$is Matrix multiplication.

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 [58][59]. 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:

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

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

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

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

$$h_t = O_t \circ \tanh(C_t), \quad (14)$$

Where $*$is the Convolution operation.

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 intrusion detection model, which must consider the spatial and temporal information of the intrusion detection 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 the discrete data, such as the connection status and the service type in the network traffic characteristics, we adopt the One-Hot encoding. For the continuous data, such as the network connection time in the characteristics of the traffic data, we normalized it in the following way:

$$x_n = \frac{x - x_{min}}{x_{max} - x_{min}}, \quad (15)$$

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 intrusion detection data by the Softmax function.

The proposed network intrusion detection model is displayed in Figure 6. 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 classificationlayer is Softmax.

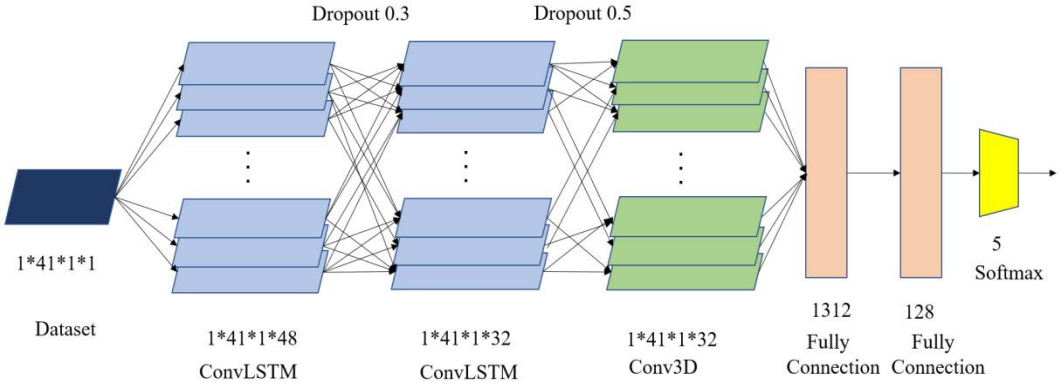

FIGURE 6. Network structure of ConvLSTM.

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 I.

TABLE I
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

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.

TABLE II
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

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. The data set is mainly used for the Knowledge Discovery and Data Mining, yet here used for the Network Intrusion. 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 III.

TABLE III
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

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

TABLE IV
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

B. Evaluation Metrics

Intrusion detection 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. Their formulas are as follows:

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

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

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3104718, IEEE Access

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:

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

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

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

In this paper, we also select the ROC curve as our metric to evaluate the intrusion detection model comprehensively. The ROC curve and AUC are usually used to evaluate the classification problem of imbalanced sample distribution, and the curve is shown in Figure 7.

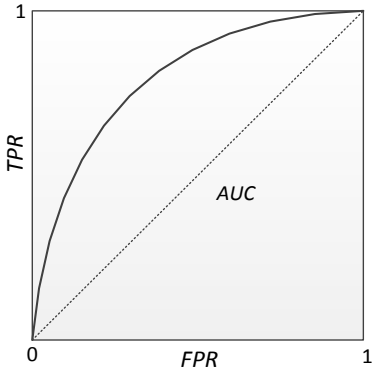

FIGURE 7. ROC cure and AUC.

Here, FPR and TPR is False Positive Rate and True Positive Rate, respectively. AUC is Area Under ROC Curve.

C. ANALYSIS AND COMPARISONS

Table V shows the prediction of all models in each category. In Table V, 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 algorithms, the ConvLSTM model is superior to other classic models in 'normal', 'dos' and 'R21'.

TABLE V
PREDICTIONOF ALL MODELS IN EACH CATEGORY

	Normal			**Dos**			**Probe**		
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
ConvLSTM	**0.723**	0.995	**0.838**	**0.999**	0.972	**0.986**	0.867	**0.713**	0.783
CNN-LSTM	0.706	0.983	0.822	0.997	0.968	0.982	0.703	0.558	0.622
CNN	0.719	**0.997**	0.835	**0.999**	**0.973**	**0.986**	0.920	0.666	0.773
LSTM	0.706	0.983	0.822	0.997	0.968	0.982	0.703	0.558	0.622
MLP	0.707	0.933	0.804	0.980	0.792	0.876	0.057	0.625	0.105
RNN	0.716	0.986	0.829	0.996	0.972	0.984	**0.925**	0.681	0.784
GRU	0.716	0.986	0.830	0.996	0.972	0.984	0.915	0.698	**0.791**
	R2l			**U2r**		
	Pre	Rec	F1	Pre	Rec	F1
ConvLSTM	**0.924**	**0.029**	**0.056**	0.833	0.022	0.043
CNN-LSTM	0.636	0.005	0.010	0	0	0
CNN	0.858	0.006	0.011	**0.900**	0.039	0.076
LSTM	0.636	0.005	0.010	0	0	0
MLP	0	0	0	0	0	0
RNN	0.396	0.001	0.002	0.684	0.057	0.105
GRU	0.590	0.003	0.006	0.710	**0.097**	**0.170**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3104718, IEEE Access

The overall weighted average prediction of all compared models is shown in Table VI. 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.

TABLE VI
WEIGHTED AVERAGE PREDICTION OF ALL MODELS IN EACH CATEGORY.

Al	WA-P	WA-R	WA-F1
ConvLSTM	**93.94%**	**92.35%**	90.49%
CNN-LSTM	91.68%	91.47%	89.49%
CNN	93.57%	92.23%	**92.20%**
LSTM	91.68%	91.47%	89.49%
MLP	86.23%	77.50%	80.51%
RNN	90.86%	91.96%	89.91%
GRU	91.92%	92.01%	89.98%

The ConvLSTM model can simultaneously obtain the spatial and temporal features of network traffic data, so the precisions and the recall rates perform well. In addition, we find that the ability of intrusion detection 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.

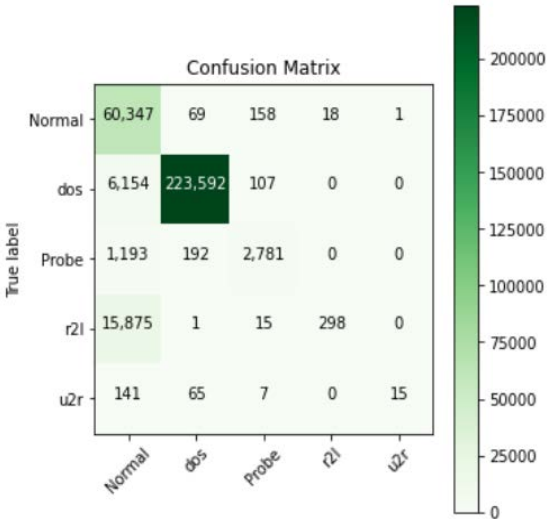

FIGURE 8. The confusion matrix of ConvLSTM.

Figure 8 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 IV. 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 9 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.

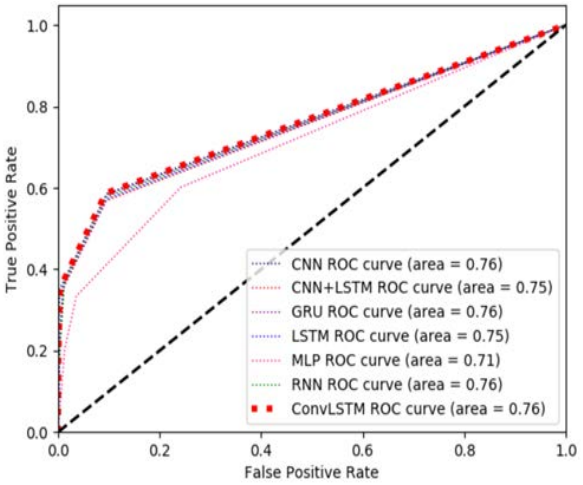

FIGURE 9. ROC curve results for all models.

In summary, in order to verify the effectiveness of the models, we select a large number of evaluation metrics, including Precision, Recall, F1-Score, Weighted Average, etc. The above-discussed results confirm that theperformance of the proposed ConvLSTM model is competitive to the state-of-the-art intrusion detection 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 intrusion detection.

V. CONCLUSION

In this paper, we propose an end-to-end trainable neural network model for the network traffic intrusion detection. In this model, the intrusion detection data is preprocessed 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. In addition, we have carried out a lot of comparative experiments from different aspects, including the overall effect of intrusion detection, the detection effect for different types of samples, the confusion matrix results for different types of samples, and the results of ROC curve. The results show that the proposed model effectively improves the intrusion detection ability of network traffic, and provides a new idea for intrusion detection 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.

FUNDING

This research was funded by “China Postdoctoral Science Foundation, grant number 2020M670998”, “National Natural Science Foundation of China, grant number U1836207,” and “National key R & D program of China, grant number 2018YFC0807105”.

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] L.Y. Shi,H.Q. Zhu., Y.H. Liu,and J. Liu,“Intrusion Detection of Industrial Control System Based on Correlation Information Entropy and CNN-BiLSTM,”Journal of Computer Research and Development, vol. 56, no. 11, pp. 2330-2338, 2019.

[2] Y.L. Ding and Y.Q. Zhai, “Intrusion detection system for NSL-KDD dataset using convolutional neural networks,” Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence,pp. 81-85, 2018.

[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 Transactions on software engineering, vol. SE-13, no. 2, pp. 222-232, 1987.

[5] M.Akbanov,V. G.Vassilakis and M. D. Logothetis, “Ransomware detection and mitigation using software-defined networking: The case of WannaCry,”Computers & Electrical Engineering,vol. 76, pp.111-121, 2019.

[6] D.Kreutz,F. M.Ramos,P. E.Verissimo,C. E.Rothenberg,S.Azodolmolky and S. Uhlig, “Software-defined networking: A comprehensive survey,”Proceedings of the IEEE, vol. 103, no.1,pp.14-76, 2014.

[7] M. K.Nanda and M. R.Patra,“ Intrusion Detection and Classification Using Decision Tree-Based Feature Selection Classifiers,” Intelligent and Cloud Computing, Springer, Singapore, pp.157-170, 2021.

[8] S. M.Badr, “Adaptive layered approach using C5. 0 decision tree for intrusion detection systems (ALIDS),” International Journal of Computer Applications, vol.66, no.22, pp. 18-22, 2013.

[9] L.Portnoy, E. Eskin,and S. Stolfo, “Intrusion detection with unlabeled data using clustering,”Acm workshop on data mining applied, 2001.

[10] A. A.Chormale and A. P. Ghatule, “Cloud Intrusion Detection System Using Fuzzy Clustering and Artificial Neural Network,” Journal of Physics: Conference Series, IOP Publishing, vol. 1478, no. 1, pp. 012030, 2020.

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

[12] V.Pai and N. D.Adesh, “Comparative analysis of Machine Learning algorithms for Intrusion Detection,” IOP Conference Series: Materials Science and Engineering, IOP Publishing, vol. 1013, no. 1, pp. 012038, 2021.

[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 Journal of the Computer Society, vol. 2, pp.14-25, 2021.

[14] G.Andresini, A. Appice and D. Malerba, “Nearest cluster-based intrusion detection through convolutional neural networks,” Knowledge-Based Systems, vol. 216, pp. 106798, 2021.

[15] H.Liu,S. Z.Peng andB.F.Luo, “Research on intrusion detection model based on ecosystem neural network,”Command Control & Simulation,” vol. 42, no.4,pp.45-50, 2020.

[16] L. Z.Kong, “Research on intrusion detection algorithm based on network anomaly,” Beijing: Beijing Jiao tong University, 2017.

[17] J.Kim and H.Kim, “Applying recurrent neural network to intrusion detection with hessian free optimization,” International Workshop on Information Security Applications, Springer, Cham, pp.357-369, 2015.

[18] S.Leyi,Z.Hongqiang,L.Yihao and L.Jia, “Intrusion Detection of Industrial Control System Based on Correlation Information Entropy and CNN-BiLSTM,” Journal of Computer Research and Development, vol. 56, no. 1,pp. 2330, 2019.

[19] E.Raff,J.Sylvester and C.Nicholas, “Learning the pe header, malware detection with minimal domain knowledge,” Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pp.121-132, 2017.

[20] X. Shi, Z. Chen, H.Wang, D. Y.Yeung,W. K.Wong andW. C.Woo, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” arXiv preprint, arXiv:1506.04214, 2015.

[21] E. E.Osuna, “Support vector machines: Training and applications,” Massachusetts Institute of Technology, 1998.

[22] J. L.Elman, “Distributed representations, simple recurrent networks, and grammatical structure,” Machine learning, vol. 7, no. 2,pp.195-225, 1991.

[23] D. E.Rumelhart,G. E.Hinton,R. J.Williams, “Learning representations by back-propagating errors”, nature, vol. 323, no.6088, pp.533-536, 1986.

[24] M. I. Jordan, “Serial order: A parallel distributed processing approach”, Advances in psychology, North-Holland, vol.121, pp.471-495, 1997.

[25] J.Schmidhuber, “Deep learning in neural networks: An overview”, Neural networks, vol. 61, pp.85-117, 2015.

[26] L. B.Almeida, “A learning rule for asynchronous perceptrons with feedback in a combinatorial environment”, Artificial neural networks: concept learning, pp.102-111, 1990.

[27] R. J.Williams and D. Zipser,“learning algorithm for continually running fully recurrent neural networks,” Neural computation, vol.1, no.2,pp.270-280, 1989.

[28] P. J.Werbos, "Backpropagation through time: what it does and how to do it," Proceedings of the IEEE, vol. 78, no. 10,pp.1550-1560, 1990.
[29] S.Hochreiter, "Untersuchungen zu dynamischen neuronalen Netzen," Diploma, Technische Universität München, 1991.
[30] J.Chung,C.Gulcehre,K.Choand Y.Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv preprint, arXiv:1412.3555, 2014.
[31] R.Lowe,N. Pow,I.Serban andJ. Pineau, "The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems," arXiv preprint, arXiv:1506.08909, 2015.
[32] R. Z. Yao,N. Wang, Z.Liu,P. Chen andX. J. Sheng, "Intrusion Detection System in the Advanced Metering Infrastructure: A Cross-Layer Feature-Fusion CNN-LSTM-Based Approach," Sensors, vol. 21, no. 2,pp. 626, 2021.
[33] J.Schmidhuber, "Learning complex, extended sequences using the principle of history compression," Neural Computation, vol. 4, no. 2,pp.234-242, 1992.
[34] S.Zhang,Y.Gong andJ.Wang, "The development of deep convolution neural network and its applications on computer vision,"Chinese Journal of Computers, 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. Journal of Computer Research and Development,"vol. 57, no. 11, pp. 2456, 2020.
[36] K.Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint, arXiv:1409.1556, 2014.
[37] K. M. He, X. Y. Zhang,S. Q. Ren,J. Sun, "Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition,"pp.770-778, 2016.
[38] J.Redmon,S. Divvala,R.Girshick andA. Farhadi, "You only look once: Unified, real-time object detection," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.779-788, 2016.
[39] J.Redmon and A.Farhadi, "YOLO9000: better, faster, stronger," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.7263-7271, 2017.
[40] J.Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018.
[41] A.Bochkovskiy,C. Y. Wang,H. Y.Liao, "Yolov4: Optimal speed and accuracy of object detection," arXiv preprint, arXiv:2004.10934, 2020.
[42] X.Glorot, A.Bordes, Y.Bengio, "Deep sparse rectifier neural networks," Proceedings of the fourteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings, 2011, pp.315-323.
[43] B.Xu,N. Wang,T. Chen,M. Li, "Empirical evaluation of rectified activations in convolutional network," arXiv preprint , arXiv:1505.00853, 2015.
[44] T.Sercu,C. Puhrsch,B. Kingsbury andY. LeCun, "Very deep multilingual convolutional neural networks for LVCSR," 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, pp.4955-4959, 2016.
[45] F.Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions,"arXiv preprint, arXiv:1511.07122, 2015.
[46] A. G.Howard,M. Zhu,B. Chen,D. Kalenichenko, W.Wang, *et al.,* "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint, arXiv:1704.04861, 2017.
[47] M.Sandler,A. Howard,M. Zhu,A. Zhmoginov,L. C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.4510-4520, 2018.
[48] A.Howard,M. Sandler,G. Chu,L. C. Chen,B. Chen,M. Tan, *et al.,* "Searching for mobilenetv3," Proceedings of the IEEE/CVF International Conference on Computer Vision, pp.1314-1324, 2019.
[49] A.Krizhevsky,I. Sutskever andG. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, no. 25, pp.1097-1105, 2012.
[50] T.Weiss,J. Hillenbrand,A. Krohn,F. K. Jondral,"Mutual interference in OFDM-based spectrum pooling systems," 2004 IEEE 59th Vehicular Technology Conference, VTC 2004-Spring (IEEE Cat. No. 04CH37514), IEEE, vol. 4, pp.1873-1877, 2004.
[51] K.He,X. Zhang,S. Ren,J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," IEEE transactions on pattern analysis and machine intelligence, vol. 37, no. 9, pp.1904-1916, 2015.
[52] C.Szegedy,W. Liu,Y. Jia,P.Sermanet,S. Reed, *et al.,* "Going deeper with convolutions," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.1-9, 2015.
[53] S.Ioffe andC.Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," International conference on machine learning, PMLR, pp.448-456, 2015.
[54] C.Szegedy,V. Vanhoucke,S. Ioffe,J. Shlens and Z. Wojna, "Rethinking the inception architecture for computer vision," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.2818-2826, 2016.
[55] M.Lin,Q. Chen andS. Yan, "Network in network," arXiv preprint, arXiv:1312.4400, 2013.
[56] S.Hochreiter andJ. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8,pp.1735-1780, 1997.
[57] F. A.Gers,J. Schmidhuber andF.Cummins, "Learning to forget: Continual prediction with LSTM,"Neural computation, vol. 12, no. 10, pp. 2451-2471, 2000.
[58] X.Shi,Z. Chen,H. Wang,D. Y. Yeung,W. K. Wong,W. C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," arXiv preprint, arXiv:1506.04214, 2015.
[59] Y.Liu,J. J. Cao andX. C. Diao, "Survey on stability of feature selection," Journal of Software, vol. 29, pp. 2559-2579, 2018.