IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 12, 2020, accepted January 31, 2020, date of publication February 10, 2020, date of current version February 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2972627

BAT: Deep Learning Methods on Network Intrusion Detection Using NSL-KDD Dataset

TONGTONG SU, HUAZHI SUN, JINQI ZHU, SHENG WANG, AND YABO LI
School of Computer and Information Engineering, Tianjin Normal University, Tianjin 300387, China

Corresponding authors: Huazhi Sun (sunhuazhi@mail.tjnu.edu.cn) and Jinqi Zhu (zhujinqi1016@163.com)

This work was supported in part by the Natural Science Foundation of Tianjin under Grant 17JCYBJC16400, Grant 18JCYBJC85900, and Grant 18JCQNJC70200, and in part by the Science and Technology Development Fund of Tianjin Education Commission for Higher Education under Grant JW1702.

ABSTRACT Intrusion detection can identify unknown attacks from network traffics and has been an effective means of network security. Nowadays, existing methods for network anomaly detection are usually based on traditional machine learning models, such as KNN, SVM, etc. Although these methods can obtain some outstanding features, they get a relatively low accuracy and rely heavily on manual design of traffic features, which has been obsolete in the age of big data. To solve the problems of low accuracy and feature engineering in intrusion detection, a traffic anomaly detection model BAT is proposed. The BAT model combines BLSTM (Bidirectional Long Short-term memory) and attention mechanism. Attention mechanism is used to screen the network flow vector composed of packet vectors generated by the BLSTM model, which can obtain the key features for network traffic classification. In addition, we adopt multiple convolutional layers to capture the local features of traffic data. As multiple convolutional layers are used to process data samples, we refer BAT model as BAT-MC. The softmax classifier is used for network traffic classification. The proposed end-to-end model does not use any feature engineering skills and can automatically learn the key features of the hierarchy. It can well describe the network traffic behavior and improve the ability of anomaly detection effectively. We test our model on a public benchmark dataset, and the experimental results demonstrate our model has better performance than other comparison methods.

INDEX TERMS Network traffic, intrusion detection, deep learning, BLSTM, attention mechanism.

I. INTRODUCTION

With the development and improvement of Internet technology, the Internet is providing various convenient services for people. However, we are also facing various security threats. Network viruses, eavesdropping and malicious attacks are on the rise, causing network security to become the focus of attention of the society and government departments. Fortunately, these problems can be well solved via intrusion detection. Intrusion detection plays an important part in ensuring network information security. However, with the explosive growth of Internet business, traffic types in the network are increasing day by day, and network behavior characteristics are becoming increasingly complex, which brings great challenges to intrusion detection [1], [2]. How to identify various malicious network traffics, especially unexpected malicious network traffics, is a key problem that cannot be avoided. In fact, network traffic can be divided into two categories (normal traffics and malicious traffics). Furthermore, network traffic can also be divided into five categories: Normal, DoS (Denial of Service attacks), R2L (Root to Local attacks), U2R (User to Root attack) and Probe (Probing attacks). Hence, intrusion detection can be considered as a classification problem. By improving the performance of classifiers in effectively identifying malicious traffics, intrusion detection accuracy can be largely improved.

Machine learning methods [3]–[8] have been widely used in intrusion detection to identify malicious traffic. However, these methods belong to shallow learning and often emphasize feature engineering and selection. They have difficulty in features selection and cannot effectively solve the massive intrusion data classification problem, which leads to low recognition accuracy and high false alarm rate. In recent years, intrusion detection methods based on deep learning have been proposed successively. In [9], the authors propose a mal-ware traffic classification method based on convolutional

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

neural network with traffic data as image. This method does not need manual design features, and directly takes the original traffic as the input data to the classifier. In [10], the authors provide an analysis of the viability of Recurrent Neural Networks (RNN) to detect the behavior of network traffic by modeling it as a sequence of states that change over time. In [11], the authors verify the performance of Long Short-Term memory (LSTM) network in classifying intrusion traffics. Experimental results show that LSTM can learn all the attack classes hidden in the training data. All the above methods treat the entire network traffic as a whole consisting of a sequence of traffic bytes. They don't make full use of domain knowledge of network traffics. For example, CNN converts continuous network traffic into images for processing, which is equivalent to treating traffics as independent and ignore the internal relations of network traffics. Firstly, network traffic is a hierarchical structure. Specifically, network traffic is a traffic unit composed of multiple data packets. Data packet is a traffic unit composed of multiple bytes. Secondly, traffic features in the same and different packets are significantly different. Sequential features between different packets need to be extracted independently. In other words, not all traffic features are equally important for traffic classification in the process of extracting features on a certain network traffic.

However, little prior works have utilized the above mentioned structure of network traffic. Inspired by these characteristics, in this paper, we propose and demonstrate our method to analyze network traffic in an overall view. Network traffic is generally collected at fixed time intervals. Repeating this collecting process for m times, we can get the network traffic X', where $X' = (x'_1, x'_2, \ldots, x'_m)$ is a matrix with m data packets. Each x represents a data packet, in data packet is seen as a whole consisting of a sequence of traffic bytes. Before entering the data into the BAT model, the original data is preprocessed by multiple convolutional layers. Global features can be obtained with the increase of the convolutional layer. With the preprocessing, we get an abstract representation of network traffic X from X'. In order to better make full use of domain knowledge of network traffics, we propose a deep learning model BAT-MC that mainly combines bidirectional long-term memory (BLSTM) [12] and attention mechanism [13]. BLSTM is used to learn the characteristics of each packet and get the vector corresponding to each packet. Attention mechanism is then used to perform feature learning on the sequence data composed of the packet vector to obtain the fine-grained features. Up to now, we have finished the key features extraction of network traffics via attention mechanism. The whole process of feature learning does not use any feature engineering skills. The automatically learnt key features can better describe the traffic behavior, which can effectively improve the anomaly detection capability. Finally, a full connected network and a *softmax* function are performed on the obtained fine-grained features for anomaly detection. To verify the effectiveness of the BAT-MC network, it is comprehensively evaluated on the NSL-KDD dataset and gets the best results. The accuracy of the BAT-MC network can reach 84.25%, which is about 4.12% and 2.96% higher than the existing CNN and RNN model, respectively.

The following are some of the key contributions and findings of our work:

1) We propose an end-to-end deep learning model BAT-MC that is composed of BLSTM and attention mechanism. BAT-MC can well solve the problem of intrusion detection and provide a new research method for intrusion detection.
2) We introduce the attention mechanism into the BLSTM model to highlight the key input. Attention mechanism conducts feature learning on sequential data composed of data package vectors. The obtained feature information is reasonable and accurate.
3) We compare the performance of BAT-MC with traditional deep learning methods, the BAT-MC model can extract information from each packet. By making full use of the structure information of network traffic, the BAT-MC model can capture features more comprehensively.
4) We evaluate our proposed network with a real NSL-KDD dataset. The experimental results show that the performance of BAT-MC is better than the traditional methods.

The rest of the paper is organized as follows: In Section 2, we give a brief overview of the related work, especially how intelligent algorithms facilitate the development of intrusion detection. In Section 3, we present details of the proposed BAT-MC model. In Section 4, we explain the experimental setup and present our results. The performance of BAT-MC model is compared with other machine learning methods both in binary classification and multiclass classification. Section 5 draws the conclusions.

II. RELATED WORKS

The intrusion detection technology can be divided into three major categories: pattern matching methods, traditional machine learning methods and deep learning methods.

At the beginning, people mainly use pattern matching algorithms for intrusion detection. Pattern matching algorithm [14], [15] is the core algorithm of intrusion detection system based on feature matching. Most algorithms have been considered for use in the past. In [16], the authors make a summary of pattern matching algorithm in Intrusion Detection System: KMP algorithm, BM algorithm, BMH algorithm, BMHS algorithm, AC algorithm and AC-BM algorithm. Experiments show that the improved algorithm can accelerate the matching speed and has a good time performance. In [17], Naive approach, Knuth-MorrisPratt algorithm and RabinKarp Algorithm are compared in order to check which of them is most efficient in pattern/intrusion detection. Pcap files have been used as datasets in order to determine the efficiency of the algorithm by taking into consideration their running times respectively. These traditional pattern

recognition algorithms have serious defects, which cannot achieve the effect of intrusion detection. Finding an efficient algorithm that reaches high efficiency and low false positive rates is still the focus of current work. With the development of artificial intelligence, the application of intelligent algorithms for intrusion detection has become a new research hotspot.

The traffic anomaly detection methods based on machine learning have achieved a lot of success. In [18], the authors propose a new method of feature selection and classification based on support vector machine (SVM). Experimental results on NSL-KDD cup 99 of intrusion detection data set showed that the classification accuracy of this method with all training features reached 99%. In [19], the authors combine k-mean clustering on the basis of KNN classifier. The experimental results on NSL-KDD dataset show that this method greatly improves the performance of KNN classifier. In [20], the authors propose a new framework to combine the misuse and the anomaly detection in which they apply the random forests algorithm. Experimental results show that the overall detection rate of the hybrid system is 94.7% and the overall false positive rate is 2%. In [21], the performance of NSL-KDD dataset is evaluated via Artificial Neural Network (ANN). The detection rate obtained is 81.2% and 79.9% for intrusion detection and attack type classification task respectively for NSL-KDD dataset. In [22], an intrusion detection method based on decision tree (DT) is proposed. Experimental results of feature selection using the relevant feature selection (CFS) subset evaluation method show that the DT based intrusion detection system has a higher accuracy. As described above, machine learning methods have been proposed and have achieved success for an intrusion detection system. However, these methods require large-scale preprocessing and complex feature engineering of traffic data. It is impossible to solve the massive intrusion data classification problem using machine learning methods.

With the superior performance of deep learning in image recognition [23], [24] and speech recognition [25], [26], traffic anomaly detection methods based on deep learning have been proposed. In [27], the authors use Self-taught Learning (STL) on NSL-KDD dataset for network intrusion. Testing results show that their 5-class classification achieved an average f-score of 75.76%. In [28], the authors propose an intrusion detection method using deep belief network (DBN) and probabilistic neural network (PNN). The experiment result on the KDD CUP 1999 dataset shows that the method performs better than the traditional PNN, PCA-PNN and unoptimized DBN-PNN. Similarly, [29] and [30] train the DBN as a classifier to detect intrusions. In [31], the authors propose a novel network intrusion detection model utilizing convolutional neural networks (CNNs). The CNN model not only reduces the false alarm rate (FAR) but also improves the accuracy of the class with small numbers. In [32], an artificial intelligence (AI) intrusion detection system using a deep neural network (DNN) is investigated and tested with the KDD Cup 99 dataset in response to ever-evolving network attacks. The results show a significantly high accuracy and detection rate, averaging 99%. However, current deep learning methods don't make full use of the structured information of network traffic. Network traffic is essentially a kind of time series data. Similar to the structure of letters, words, sentences and paragraphs in natural language processing (NLP), network traffic is composed of multiple data packets and each data packet is a set of multiple bytes.

In this paper, drawing on the application methods of deep learning in NLP, we adopt phased processing. The BLSTM is used to learn the sequential features in the data packet to obtain a vector corresponding to each data packet. Then, attention layer is used to perform feature learning on the sequential data composed of the packet vector. Attention can filter out the characteristics to get a network flow vector, which are helpful to achieve more accurate network traffic classification. Through the learning of two phases of BLSTM and attention on the time series features, the BAT-MC model finally outputs a network flow vector, which contains structured information of network traffic. Hence, the BAT-MC model makes full use of the structure information of network traffic.

III. PROPOSED WORK

As shown in Figure 1, the BAT-MC model consists of five components, including the input layer, multiple convolutional Layers, BSLTM layer, attention layer and output layer, from bottom to top. At the input layer, BAT-MC model converts each traffic byte into a one-hot data format. Each traffic byte is encoded as an n-dimensional vector. After traffic byte is converted into a numerical form, we perform normalization operations. At the multiple convolutional layer, we convert the numerical data into traffic images. Convolutional operation is used as a feature extractor that takes an image representation of data packet. At the BLSTM layer, BLSTM model which connects the forward LSTM and the backward LSTM is used to extract features on the the traffic bytes of each packet. BLSTM model can learn the sequential characteristics within the traffic bytes because BLSTM is suitable to the structure of network traffic. In the attention layer, attention mechanism is used to analyze the important degree of packet vectors to obtain fine-grained features which are more salient for malicious traffic detection. At the output layer, the features generated by attention mechanism are then imported into a fully connected layer for feature fusion, which obtains the key features that accurately characterize network traffic behavior. Finally, the fused features are fed into a classifier to get the final recognition results.

A. DATA PREPROCESSING LAYER

There are three symbolic data types in NSL-KDD data features: protocol type, flag and service. We use one-hot encoder mapping these features into binary vectors.

One-Hot Processing: NSL-KDD dataset is processed by one-hot method to transform symbolic features into numerical features. For example, the second feature of the

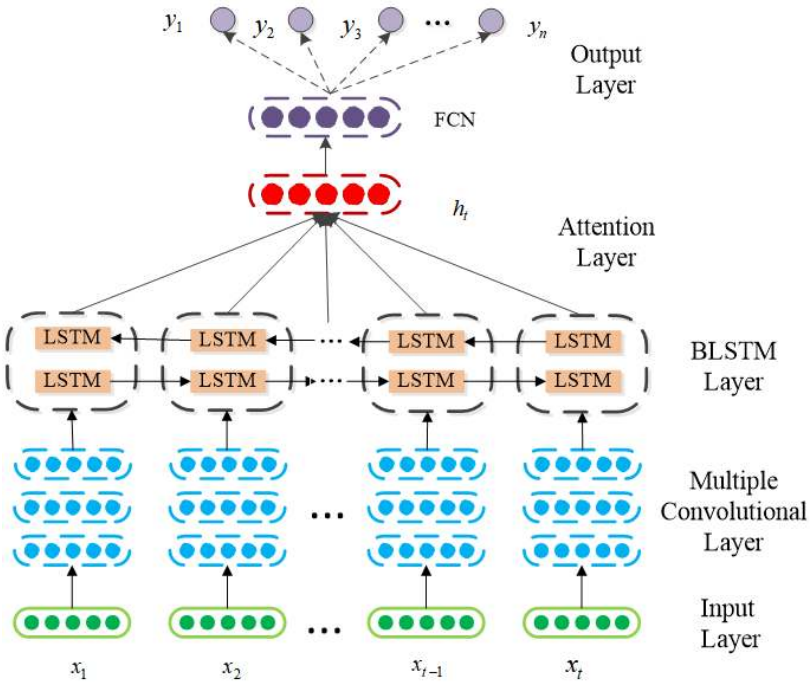

FIGURE 1. The Architecture of BAT-MC model. The whole architecture is divided into five parts.

NSL-KDD data sample is protocol type. The protocol type has three values: tcp, udp, and icmp. One-hot method is processed into a binary code that can be recognized by a computer, where tcp is [1, 0, 0], udp is [0, 1, 0], and icmp is [0, 0, 1].

Normalization Processing: The value of the original data may be too large, resulting in problems such as "large numbers to eat decimals", data processing overflows, and inconsistent weights so on. We use standard scaler to normalize the continuous data into the range [0, 1]. Normalization processing eliminates the influence of the measurement unit on the model training, and makes the training result more dependent on the characteristics of the data itself. The formula is shown in equation (1) and equation (2).

$$r' = \frac{r - r_{min}}{r_{max} - r_{min}}, \tag{1}$$

$$r_{max} = max\{r\}, \tag{2}$$

where r stands for numeric feature value, r_{min} stands for the minimal value of the feature, r_{max} stands for the max value, r' stands the value after the normalization.

B. MULTIPLE CONVOLUTIONAL LAYERS

After the above processing operations, convolutional layer is used to capture the local features of traffic data. Convolutional layer [33], [34] is the most important part of the CNN, which convolves the input images (or feature maps) with multiple convolutional kernels to create different feature maps. According to [35], the shallower convolutional layers whose receptive field is narrow can extract local information, and while the deeper layers can capture global information with larger vision field. Hence, as the number of the convolutional layers increases, the scale of the convolutional feature gradually becomes coarser. In this paper, the input of the convolutional layer can be formulated as a tensor of the size $H \times W \times 1$, where H and W denote the height and width of data yielded by normalization processing. Suppose we have some N unites layer as input which is followed by convolutional layer. If we use m width filter w, the convolutional output will be $(N - m + 1)$ unites. The convolutional calculation process is as shown in equation (3).

$$x_{i,k}^{l,j} = f(b_j + \sum_{a=1}^{m} w_{a,k}^{j} r_{i+(k-1)\times s+a-1}^{l-1,j}), \tag{3}$$

where $x_{i,k}^{l,j}$ is one of the ith unit of j feature map of the kth section in the lth layer, and s is the range of section. f is a non-linear mapping, it usually uses hyperbolic tangent function, $tanh(\cdot)$.

C. BLSTM LAYER

For the time series data composed of traffic bytes, BLSTM can effectively use the context information of data for feature learning. The BLSTM is used to learn the time series feature in the data packet. Traffic bytes of each data packet are sequentially input into an BLSTM, which finally obtain a packet vector. BLSTM is an enhanced version of LSTM (Long Short-Term Memory) [36], [37]. The BLSTM model is used to extract coarse-grained features by connecting forward LSTM and backward LSTM. LSTM is designed by the input

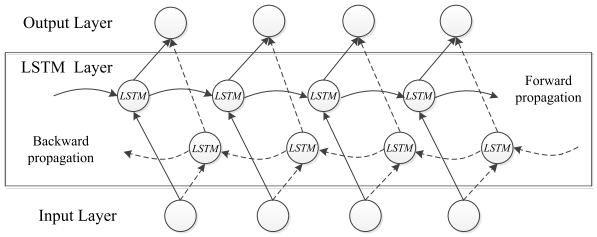

FIGURE 2. The architecture of BLSTM model.

gate i, the forget gate f and the output gate o to control how to overwrite the information by comparing the inner memory cell C when new information arrives [38]. When information enters a LSTM network, we can judge whether it is useful according to relevant rules. Only the information that meets algorithms authentication will be remained, and inconsistent information will be forgotten through forget gate. Given an input sequence $x = (x_0, \ldots, x_t)$ at time t and the hidden states of a BLSTM layer, $h = (h_0, \ldots, h_t)$ can be derived as follows.

The forget gate will take the output of hidden layer h_{t-1} at the previous moment and the input x_t at the current moment as input to selectively forget in the cell state C_t, which can be expressed as:

$$f_t = sigmoid(W_{xf}x_t + W_{hf}h_{t-1} + b_f), \tag{4}$$

The input gate cooperates with a *tanh* function together to control the addition of new information. *tanh* generates a new candidate vector. The input gate generates a value for each item in $\widetilde{C}_t$ from 0 to 1 to control how much new information will be added, which can be expressed as:

$$C_t = sigmoid(f_t \cdot C_{t-1} + i_t \cdot \widetilde{C}_t), \tag{5}$$
$$i_t = sigmoid(W_{xi}x_t + W_{hi}h_{t-1} + b_t), \tag{6}$$
$$\widetilde{C}_t = \tanh(W_c x_t + W_c h_{t-1} + b_c), \tag{7}$$

The output gate is used to control how much of the current unit state will be filtered out, which can be expressed as:

$$o_t = sigmoid(W_{xo}x_t + W_{ho}h_{t-1} + b_o), \tag{8}$$

For the BLSTM model at time t, the hidden states of the h_t that is a packet vector generated from each packet can be defined as the concatenation of $\overleftarrow{h}_t$ and $\overrightarrow{h}_t$, which can be expressed as:

$$h_t = \overleftarrow{h}_t + \overrightarrow{h}_t, \tag{9}$$
$$\overrightarrow{h}_t = \tanh(W_{x\overrightarrow{h}}x_t + W_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}), \tag{10}$$
$$\overleftarrow{h}_t = \tanh(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}), \tag{11}$$

where $'.'$ means the pointwise product. x represents the input of the heterogeneous time series data. $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ is the hidden states of forward LSTM layer and backward LSTM layer at time t. All the matrices W are the connection weights between two units, and b are bias vectors.

D. ATTENTION LAYER

BLSTM eventually generates a packet vector for each packet. These packet vectors are arranged in the order of interaction between the two parties in the network stream to form a sequence of packet vectors. The relationships within packet vectors will be learned by attention layer. similarly to [39], attention mechanism is used to adjust probability of packet vectors so that our model pays more attention to important features. Firstly, the packet vectors h_t extracted by the BLSTM model is used to obtain its implicit representation u_t through a nonlinear transformation, which can be expressed as:

$$u_t = \tanh(W_w h_t + b_w), \tag{12}$$

We next measure the importance of packet vectors based on the similarity representation u_t with a context vector u_w and obtain the normalized importance weight coefficient α_t. u_w is a random initialization matrix that can focus on important information over u_t. The weight coefficient for the above coarse-grained features can be expressed as:

$$\alpha_t = \frac{\exp(u_t^T u_w)}{\sum \exp(u_t^T u_w)}, \tag{13}$$

Finally, the fine grained feature s can be computed via the weighted sum of h_t based on α_t. s can be expressed as:

$$s = \sum \alpha_t h_t, \tag{14}$$

The fine-grained feature vector s generated from the attention mechanism is used for malicious traffic recognition with a *softmax* classifier, which can be expressed as:

$$y = softmax(W_h s + b_h), \tag{15}$$

where W_h represents the weight matrix of the classifier, which can map s to a new vector with length h. h is the number of categories of network traffics.

E. MODEL TRAINING

Training the proposed network contains a forward pass and a backward pass.

Forward Propagation The BAT-MC model is mainly composed of BLSTM layer and attention layer, each of which presents different structures and thus plays different role in the whole model. The forward propagation [40], [41] is conducted from BLSTM layer to attention layer. The input of current model is obtained by the processing of the previous model. After the completion of forward propagation, the final recognition result is obtained. The NSL-KDD dataset is defined as X. The divided training dataset and testing dataset can be expressed as x_1,x_2,x_3. After one-hot operation and normalization operation, every samples is converted into a format X'' that can be acceptable to the BAT-MC model. Meanwhile, we set the cell state vector size as S_{state}. In summary, the abnormal traffic detection algorithm based on the BAT-MC model is summarized as Algorithm 1. The objective function of our model is the cross-entropy based cost

Algorithm 1 BAT-MC Intrusion Detection Algorithms

Input: NSL-KDD dataset, adam, lr, batch_size
Output: Accuracy

get X=(x_1,x_2,x_3) from NSL-KDD dataset;
x'_1,x'_{12},x'_3 = one-hot(x_1,x_2,x_3);
x''_1, x''_{12}, x''_3 = normalization(x'_1, x'_{12}, x'_3);
conduct convolutional processing;
for $t = 1$; $t \leq T$; **do**
    create $\overleftarrow{LSTM}_{cell}$ by S_{state};
    create $\overrightarrow{LSTM}_{cell}$ by S_{state};
    connect $BLSTM_{net}$ by $\overleftarrow{LSTM}_{cell}$ and $\overrightarrow{LSTM}_{cell}$;
    initialize $BLSTM_{net}$ by seed;
    get hidden states h_t of the $BLSTM_{net}$;
end
add a full connection layer, whose value is 320;
add a dropout, whose value is 0.1;
for *each hidden state in 1:*h_t*;* **do**
    obtain h_t implicit representation u_t through a nonlinear transformation;
    generate a random initialization matrix u_w;
    obtain the normalized importance weight coefficient α_t;
    get the fine-grained feature s via α_t and h_t;
end
add a full connection layer, whose value is 1024;
add a full connection layer, whose value is 10;
P = $BAT - MC_{net}$ (X'') ;
get Loss by p_i and y_i ;
update $BAT - MC_{net}$ by Adam with loss and η
return *accuracy*, $f1 - score$;

function [42]. The goal of training this model is to minimize the cross entropy of the expected and actual outputs for all activities. The formula is shown in (16):

$$C = -\sum_i \sum_j y_i^j \ln a_i^j + (1 - y_i^j) \ln(1 - a_i^j), \qquad (16)$$

where i is the index of network traffic. j is the traffic category. a is the actual category of network traffic and y is the predicted category.

Backward Propagation: The model is trained with adam [43]. Adam is calculated by the back-propagation algorithm. Error differentials are back-propagated with the forward-backward algorithm. Back-Propagation Through Time (BPTT) [44], [45] is applied to calculate the error differentials. In this paper, we use the Back Propagation Through Time (BPTT) algorithm to obtain the derivatives of the objective function with respect to all the weights, and minimize the objective function by stochastic gradient descent.

IV. EVALUATION

In this section, we first determine the parameters of BAT-MC to obtain the optimal model through experiments which carry out on a public dataset: the NSL-KDD dataset [46], [47]. Then, we analyze the performance of the BAT-MC model. Finally, in order to verify the advancement and practicability of the BAT-MC model, we compare the performance of this model with some state-of-the-art works.

TABLE 1. Different classifications in the NSL-KDD dataset.

	Total	Normal	Dos	Probe	R2L	U2L
KDDTrain+	125973	67343	45927	11656	995	52
KDDTest+	22544	9711	7458	2421	2754	200
KDDTest-21	11850	2152	4342	2402	2754	200

A. BENCHMARK DATASETS

The final result of network traffic anomaly detection is closely related to the dataset. The NSL-KDD dataset is an enhanced version of KDD cup 1999 dataset [48], [49], which is widely used in intrusion detection experiments. The NSL-KDD dataset not only effectively solves the inherent redundant records problems of the KDD Cup 1999 dataset but also makes the number of records reasonable in the training dataset and testing dataset. The NSL-KDD dataset is mainly composed of KDDTrain+ training dataset, KDDTest+ and KDDTest-21 testing dataset, which can make a reasonable comparison with different methods of the experimental results. As shown in Table 1, the NSL-KDD dataset have different normal records and four different types of abnormal records. The KDDTest-21 dataset is a subset of the KDDTest+ and is more difficult for classification.

Network traffic is generally collected at fixed time intervals. Essentially, network traffic data is a kind of time series data. Network traffic is a traffic unit composed of multiple data packets. Each data packet is seen as a whole consisting of a sequence of traffic bytes. There are 41 features from different data packet and 1 class label for every data packet. It can be described in the following form: x = (b_0, . . . , b_i,..). b_i is the i-th feature in a data packet, and x represents a continuous features of data packet. These features include basic features (1-10), content features (11-22) and traffic features (23-41) [50]. According to its characteristics, there are four types of attacks in this dataset: DoS (Denial of Service attacks), R2L (Root to Local attacks), U2R (User to Root attack), and Probe (Probing attacks).

B. EVALUATION METRIC

In this paper, Accuracy (A) is used to evaluate the BAT-MC model. Except for accuracy, false positive rate (TPR) and false positive rate (FPR) are also introduced [51]. These three indicators are commonly used in the research field of network traffic anomaly detection, which the calculation formula is shown as follows. Where *True Positive (TP)* represents the correct classification of the Intruder. *False Positive (FP)* represents the incorrect classification of a normal user taken as an intruder. *True Negative (NP)* represents a normal

user classified correctly. *False Negative (FN)* represents an instance where the intruder is incorrectly classified as a normal user.

Accuracy represents the proportion of correctly classified samples to the total number of samples. The evaluation metric are defined as follows:

$$accuracy, A = \frac{TP + TN}{TP + FP + FN + TN}. \tag{17}$$

True Positive Rate (TPR): as the equivalent of the Detection Rate (DR), it represents the percentage of the number of records correctly identified over the total number of anomaly records.

$$DR = TPR = \frac{TP}{TP + FN}. \tag{18}$$

False Positive Rate (FPR) represents the percentage of the number of records rejected incorrectly is divided by the total number of normal records. The evaluation metric are defined as follows:

$$FPR = \frac{FP}{FP + TN}. \tag{19}$$

C. EXPERIMENTAL SETTINGS

In order to test the performance of BAT-MC model proposed in this paper, NSL-KDD dataset is used for verification. The data samples of the NSL-KDD dataset are divided into two parts: one is used to build a classifier, that is called the training dataset. The other is used to evaluate the classifier, that is called the testing dataset. There are 125,973 records in the training set and 22,543 records in the testing set. Table 2 shows the distribution of training and testing records for the (normal/attack) type of network traffic.

TABLE 2. Distribution of training and testing records.

	normal	Dos	Pbobe	U2R	R2L	Total
Train	67,343	45,927	11,656	52	995	125,973
Test	9,711	7,458	2,421	200	2,754	22,543

The operating environment of all experiments is Keras with tensorflow as the backend; Operating system is 64-bit CtOS7; Processor is E5-2620 v4; Main frequency is 2.10GHz; Memory is 32.0G; Python version is 3.6. In view of many hyper parameters existing in the BAT-MC model, we performed 100 iterations of training on the NSL-KDD set. The hyper parameters with the highest accuracy is selected as the model parameter. The BAT-MC model is also verified on the testing dataset. After lots of experiments, three one-dimensional convolution layers are adopted when building the BAT-MC model for intrusion detection task. The parameter list of BAT-MC network is set as shown in Table 3.

D. PERFORMANCE ANALYSIS OF BAT-MC

Experiments have been designed to study the performance of the BAT-MC model for 2-category and 5-category classification, such as Normal, DoS, R2L, U2R and Probe.

TABLE 3. Super parameters of the end-to-end learning model.

parameters	Filters/neurons
conv+tanh	20
conv+tanh	40
conv+tanh	60
BLSTM hidden nodes	80
BLSTM activation function	relu
Dense	320
Dropout	0.1
softmax	10
cost function	cross entropy
optimizer	Adam
batch size	128
learning rate	0.001

In the experiment of identifying malicious traffics, when there are 80 hidden nodes in the BAT-MC model, the accuracy of BAT-MC on the KDDTest+ dataset is higher. Meanwhile, the learning rate is set to 0.01 and the number of training is 100 epoches. The confusion matrix generated by the BAT-MC model on the KDDTest+ dataset is shown in Figure 3 and Figure 4. Figure 3 and Figure 4 represent the experimental results of the BAT-MC model for the 2-class and 5-class classification, respectively. The experimental results show that most samples is concentrated on the diagonal of the confusion matrix, indicating that the overall classification performance is very high. However, it can be intuitively seen from the confusion matrix in Figure 3 show that the BAT-MC network achieves good detection performance in distinguishing normal traffics from attack traffics (only 51 samples are false positives), but there is still further improvement in

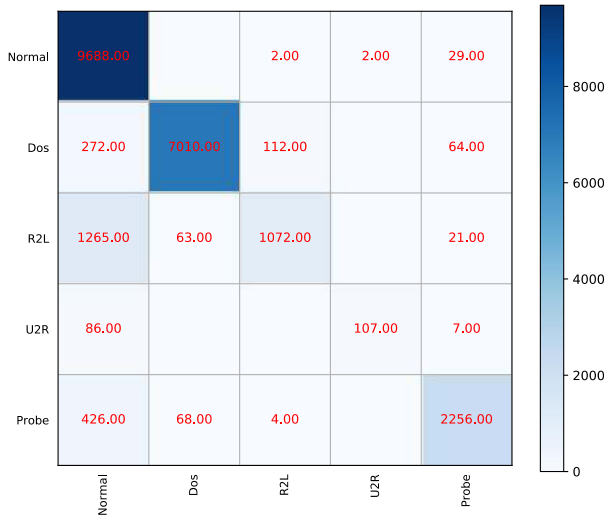

FIGURE 3. Confusion matrix yielded by the BAT-MC model (5-class).

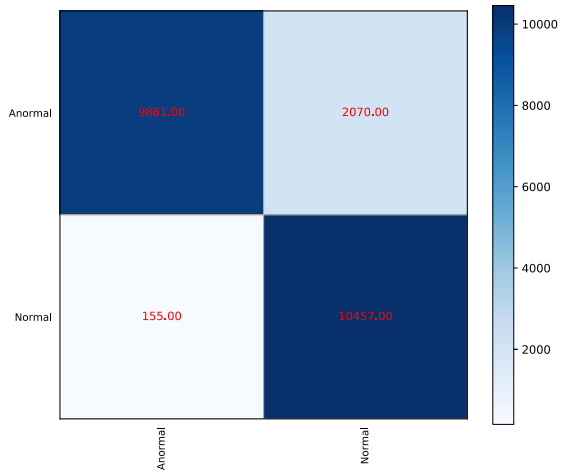

FIGURE 4. Confusion matrix yielded by the BAT-MC model (2-class).

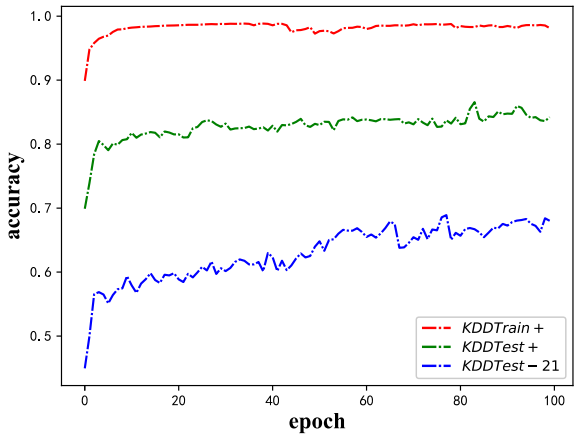

FIGURE 5. Accuracy on the KDDTest+ and KDDTest21 datasets (5-class).

distinguishing different attack traffics. The detection effect of Dos and Probe attack traffics are relatively good, while R2L and U2R attack traffics are invalid.

After careful fine-tuning, the accuracy comparison of the BAT-MC model on the KDDTest+ and KDDTest-21 set is shown in Figure 5. As the number of iterations increases, the accuracy of the BAT-MC model on both the training set and the test set shows an overall upward trend. Experiments on the KDDTest+ dataset show that when epoch = 100, the BAT-MC model has a good accuracy (84.25%). At the same time, the accuracy of the BAT-MC model on the KDDTest-21 data set is 69.42% and the accuracy on the KDDTrain+ data set is 99.21%. Table 4 shows detection rate (DR) and false positive rate (FPR) for different attack types, the motivation of intrusion detection is to obtain a higher accuracy and detection rate with a lower false positive rate. It can be seen that U2R class has the lowest detection rate and false positive rate. The U2R class with fewer samples are more likely to be misclassified than those with more samples.

Here, we evaluate the performance of our model to convolutional layer diversity. We perform the classification task on

TABLE 4. DR and FPR of the BAT-MC model on the NSL-KDD dataset (5-class).

	FPR	DR
Normal	25.70%	97.50%
Dos	1.52%	87.55%
R2L	0.91%	44.25%
U2R	0.09%	20.95%
Probe	1.15%	85.76%

TABLE 5. Convolutional layer Diversity.

layer	1	2	3	4
Accuracy	82.97%	83.91%	84.25%	84.15%

different number of convolutional layer. As shown in Table 5, the accuracy has a relatively increase when the convolutional layer increases. When the BAT-MC model does not conclude convolutional layers, the accuracy of BAT-MC reaches to 84.25%. Overall, our BAT-MC model shows a better classification accuracy (84.25%) for diverse convolutional layer.

E. COMPARISON TO THE STATE OF THE ART

In order to objectively evaluate the accuracy and differentiation of the BAT-MC network, we compare our network with some related works proposed by [52]–[54]. In [52], the authors propose a deep learning approach for intrusion detection using recurrent neural networks (RNN). Compared with traditional classification methods, such as J48, naive bayesian, and random forest, the performance obtains a higher accuracy rate and detection rate with a low false positive rate, especially under the task of multiclass classification on the NSL-KDD dataset. In [53], the authors build a Deep Neural Network (DNN) model for an intrusion detection system and train the model with the NSL-KDD Dataset. Experimental results confirm that the deep learning approach shows strong potential to be used for flow-based anomaly detection in SDN environments. In [54], the authors propose to use a typical deep learning method Convolution Neural Networks (CNN) for detecting cyber intrusions. The experimental results show that the performance of this IDS model is superior to the performance of models based on traditional machine learning methods and novel deep learning methods in multi-class classification. These works use the same dataset NSL-KDD for network traffic classification. They are not only recent highly relative and representative works on intrusion detection, but also can achieve excellent accuracy. The comparison results among these works on the NSL-KDD dataset are shown in Figure 6 and Figure 7, respectively.

As shown in Figure 6 and Figure 7, we can observe that the BAT-MC model performs better than other models in terms of accuracy, which can reach 84.25%, 69.42% in the KDDTest+ and KDDTest-21 testing set. Compared with the

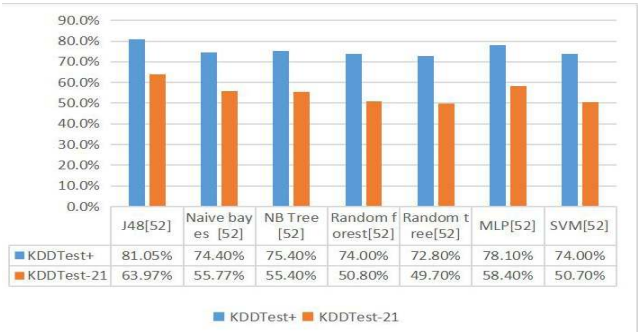

FIGURE 6. Performance of BAT-MC model and other machine learning models.

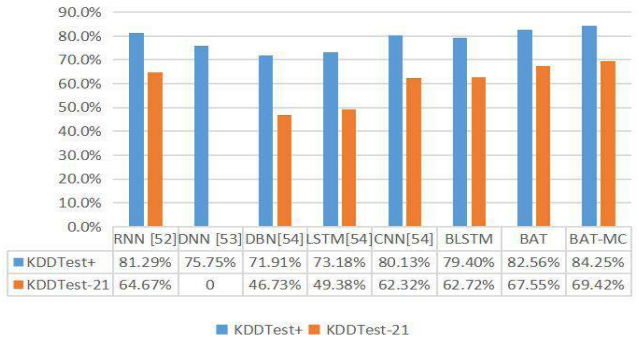

FIGURE 7. Performance of BAT-MC model and other deep learning models.

model of [52], the authors adopt the traditional machine learning methods to detect abnormal traffics. That is to say, it needs to manually design traffic features and complete the extraction and selection of network traffics before model training. In contrast, the BAT-MC model directly takes the collected traffic as original input. Then, attention mechanism captures key features from the outputs produced by the BLSTM model. Experimental results show that the BAT-MC model can automatically extract features by means of end-to-end learning, which achieves better classification results than manual design methods. Meanwhile, we compares the recent works of using deep learning model for abnormal traffic detection. As can be seen from Figure 7, the BAT-MC model achieved the best results on both the KDDTest+ and KDDTest-21 testing set. On the KDDTest+ set, the accuracy of the BAT-MC model is 4.12% and 2.96% higher than CNN [54] and RNN [52], respectively. On the KDDTest-21 set, the accuracy of the BAT-MC model is 4.75% and 7.1% higher than CNN [54] and RNN [52], respectively. The BAT-MC network is more accurate than CNN because CNN is more suitable for processing image data. Additionally, CNN uses a fixed convolution kernel that cannot model longer contextual information, which is not conducive to the feature extraction of the time series data. The BAT-MC network is better than RNN, LSTM and BLSTM because the BAT-MC model combines attention mechanism to capture the key features and obtain more context information. The BAT-MC model can capture features of network traffics more comprehensively,

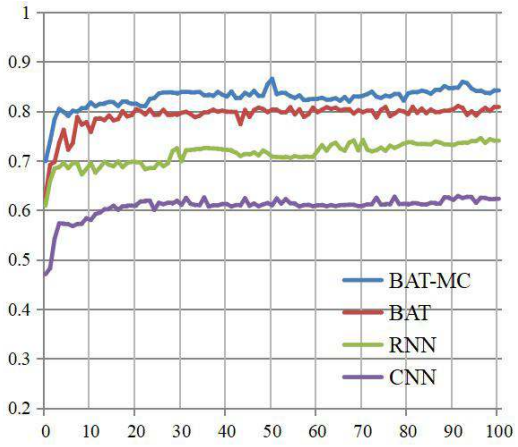

FIGURE 8. Comparison of Accuracy with different models.

which can extract the information of each data packet and then utilize it on a frame-by-frame way. These results prove that the BAT-MC network can offer a significant advantage across very different scenarios.

As the number of iterations increases, the accuracy of each model shows an overall upward trend. It can be seen from Figure 8 that the accuracy rate of testing dataset based on the BAT-MC model is not only the fastest, but the accuracy is less fluctuating after the iteration of 20 times. The accuracy of the BAT-MC model remains almost unchanged. As the number of iterations increases, the accuracy of the BAT model continues to increase, eventually reaching an ideal state. The accuracy of BAT-MC model is higher than BAT model because BAT-MC can capture global information, which proves the advantages of multiple convolutional layers. The RNN model has small-scale fluctuations in the accuracy of the iterative process. The RNN model improves faster and also has lower accuracy than BAT and BAT-MC model. The CNN model starts to improve at a slower rate and has the worst performance in each model. In summary, the BAT-MC network can accurately identify the time series data by 84.25% accuracy, which is an effective intrusion detection method.

V. CONCLUSION

The current deep learning methods in the network traffic classification research don't make full use of the network traffic structured information. Drawing on the application methods of deep learning in the field of natural language processing, we propose a novel model BAT-MC via the two phase's learning of BLSTM and attention on the time series features for intrusion detection using NSL-KDD dataset. BLSTM layer which connects the forward LSTM and the backward LSTM is used to extract features on the the traffic bytes of each packet. Each data packet can produce a packet vector. These packet vectors are arranged to form a network flow vector. Attention layer is used to perform feature learning on the network flow vector composed of packet vectors. The above feature learning process is automatically completed by deep neural network without any feature engineering technology.

This model effectively avoids the problem of manual design features. Performance of the BAT-MC method is tested by KDDTest+ and KDDTest-21 dataset. Experimental results on the NSL-KDD dataset indicate that the BAT-MC model achieves pretty high accuracy. By comparing with some standard classifier, these comparisons show that BAT-MC models results are very promising when compared to other current deep learning-based methods. Hence, we believe that the proposed method is a powerful tool for the intrusion detection problem.

REFERENCES

[1] B. B. Zarpelo, R. S Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *J. Netw. Comput. Appl.*, vol. 84, pp. 25–37, Apr. 2017.

[2] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, "Network intrusion detection," *IEEE Netw.*, vol. 8, no. 3, pp. 26–41, May 1994.

[3] S. Kishorwagh, V. K. Pachghare, and S. R. Kolhe, "Survey on intrusion detection system using machine learning techniques," *Int. J. Control Automat.*, vol. 78, no. 16, pp. 30–37, Sep. 2013.

[4] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," *Peer-to-Peer Netw. Appl.*, vol. 12, no. 2, pp. 493–501, Mar. 2019.

[5] M. Panda, A. Abraham, S. Das, and M. R. Patra, "Network intrusion detection system: A machine learning approach," *Intell. Decis. Technol.*, vol. 5, no. 4, pp. 347–356, 2011.

[6] W. Li, P. Yi, Y. Wu, L. Pan, and J. Li, "A new intrusion detection system based on KNN classification algorithm in wireless sensor network," *J. Electr. Comput. Eng.*, vol. 2014, pp. 1–8, Jun. 2014.

[7] S. Garg and S. Batra, "A novel ensembled technique for anomaly detection," *Int. J. Commun. Syst.*, vol. 30, no. 11, p. e3248, Jul. 2017.

[8] F. Kuang, W. Xu, and S. Zhang, "A novel hybrid KPCA and SVM with GA model for intrusion detection," *Appl. Soft Comput..*, vol. 18, pp. 178–184, May 2014.

[9] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, 2017, pp. 712–717.

[10] P. Torres, C. Catania, S. Garcia, and C. G. Garino, "An analysis of Recurrent Neural Networks for Botnet detection behavior," in *Proc. IEEE Biennial Congr. Argentina (ARGENCON)*, Jun. 2016, pp. 1–6.

[11] R. C. Staudemeyer and C. W. Omlin, "ACM press the south African institute for computer scientists and information technologists conference - east London, south Africa (2013.10.07-2013.10.09) proceedings of the south African institute for computer scientists and information technologists co," in *Proc. South African Inst. Comput. Scientists Inf. Technol. Conf.*, 2013, pp. 252–261.

[12] S. Cornegruta, R. Bakewell, S. Withey, and G. Montana, "Modelling radiological language with bidirectional long short-term memory networks," in *Proc. 7th Int. Workshop Health Text Mining Inf. Anal.*, 2016, pp. 1–11.

[13] O. Firat, K. Cho, and Y. Bengio, "Multi-way, multilingual neural machine translation with a shared attention mechanism," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 1–10.

[14] H. Zhang, "Design of intrusion detection system based on a new pattern matching algorithm," in *Proc. Int. Conf. Comput. Eng. Technol.*, Jan. 2009, pp. 545–548.

[15] C. Yin, "An improved BM pattern matching algorithm in intrusion detection system," *Appl. Mech. Mater.*, vols. 148–149, pp. 1145–1148, Jan. 2012.

[16] P.-F. Wu and H.-J. Shen, "The research and amelioration of pattern-matching algorithm in intrusion detection system," in *Proc. IEEE 14th Int. Conf. High Perform. Comput. Commun., IEEE 9th Int. Conf. Embedded Softw. Syst.*, Jun. 2012, pp. 1712–1715.

[17] V. Dagar, V. Prakash, and T. Bhatia, "Analysis of pattern matching algorithms in network intrusion detection systems," in *Proc. 2nd Int. Conf. Adv. Comput., Commun., Autom. (ICACCA)*, Sep. 2016, pp. 1–5.

[18] M. S. Pervez and D. M. Farid, "Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs," in *Proc. 8th Int. Conf. Softw., Knowl., Inf. Manage. Appl. (SKIMA*, Dec. 2014, pp. 1–6.

[19] H. Shapoorifard and P. Shamsinejad, "Intrusion detection using a novel hybrid method incorporating an improved KNN," *Int. J. Control Automat.*, vol. 173, no. 1, pp. 5–9, Sep. 2017.

[20] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 5, pp. 649–659, Sep. 2008.

[21] B. Ingre and A. Yadav, "2015 international conference on signal processing and communication engineering systems (spaces)," in *Proc. Int. Conf. Signal Process. Commun. Eng. Syst.*, 2015, pp. 1–15.

[22] B. Ingre, A. Yadav, and A. K. Soni, "Decision tree based intrusion detection system for NSL-KDD dataset," in *Proc. Int. Conf. Inf. Commun. Technol. Intell. Syst.*, 2017, pp. 207–218.

[23] M. Asadi-Aghbolaghi, A. Clapes, M. Bellantonio, H. J. Escalante, V. Ponce-Lopez, X. Baro, I. Guyon, S. Kasaei, and S. Escalera, "A survey on deep learning based approaches for action and gesture recognition in image sequences," in *Proc. 12th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May 2017, pp. 476–483.

[24] Z. Yan, "Multi-instance multi-stage deep learning for medical image recognition," *Deep Learn. Med. Image Anal.*, pp. 83–104, Jan. 2017.

[25] Z. Zhang, J. Geiger, J. Pohjalainen, A. E.-D. Mousa, W. Jin, and B. Schuller, "Deep learning for environmentally robust speech recognition," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 5, pp. 1–28, 2017.

[26] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata, "Audio-visual speech recognition using deep learning," *Appl. Intell.*, vol. 42, no. 4, pp. 722–737, Jun. 2015.

[27] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in *Proc. 9th EAI Int. Conf. Bio-Inspired Inf. Commun. Technol. (BIONETICS)*, 2016, pp. 21–26.

[28] G. Zhao, C. Zhang, and L. Zheng, "Intrusion detection using deep belief network and probabilistic neural network," in *Proc. IEEE Int. Conf. Comput. Sci. Eng. (CSE), IEEE Int. Conf. Embedded Ubiquitous Comput. (EUC)*, Jul. 2017, pp. 639–642.

[29] N. Gao, L. Gao, Q. Gao, and H. Wang, "An intrusion detection model based on deep belief networks," in *Proc. 2nd Int. Conf. Adv. Cloud Big Data*, Nov. 2014, pp. 247–252.

[30] M. Z. Alom, V. Bontupalli, and T. M. Taha, "Intrusion detection using deep belief networks," in *Proc. Nat. Aerosp. Electron. Conf. (NAECON)*, Jun. 2015, pp. 247–252.

[31] K. Wu, Z. Chen, and W. Li, "A novel intrusion detection model for a massive network using convolutional neural networks," *IEEE Access*, vol. 6, pp. 50850–50859, 2018.

[32] J. Kim, N. Shin, S. Y. Jo, and S. Hyun Kim, "Method of intrusion detection using deep neural network," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2017, pp. 313–316.

[33] A. Tatsuma and M. Aono, "Food image recognition using covariance of convolutional layer feature maps," *IEICE Trans. Inf. Syst.*, vol. E99.D, no. 6, pp. 1711–1715, 2016.

[34] Z. Yu, T. Li, G. Luo, H. Fujita, N. Yu, and Y. Pan, "Convolutional networks with cross-layer neurons for image recognition," *Inf. Sci.*, vols. 433–434, pp. 241–254, Apr. 2018.

[35] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4898–4906.

[36] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[37] F. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, Jan. 2016.

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

[39] N. Pappas and A. Popescu-Belis, "Multilingual hierarchical attention networks for document classification," in *Proc. IJCNLP*, 2017, pp. 1–11.

[40] Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu, and H. Zhang, "Deep learning with long short-term memory for time series prediction," *IEEE Commun. Mag.*, vol. 57, no. 6, pp. 114–119, Jun. 2019.

[41] S. Iamsa-at and P. Horata, "Handwritten character recognition using histograms of oriented gradient features in deep learning of artificial neural network," in *Proc. Int. Conf. IT Converg. Secur. (ICITCS)*, Dec. 2013.

[42] A. Boubezoul and S. Paris, "Application of global optimization methods to model and feature selection," *Pattern Recognit.*, vol. 45, no. 10, pp. 3676–3686, Oct. 2012.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980
[44] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang, "Convolutional neural networks for human activity recognition using mobile sensors," in *Proc. 6th Int. Conf. Mobile Comput., Appl. Services*, 2014, pp. 197–205.
[45] A. Graves, S. FernÍćndez, and F. Gomez, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2006, pp. 369–376.
[46] S. Revathi and A. Malathi, "A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection," *Int. J. Eng. Res. Technol.*, vol. 2, no. 12, pp. 1848–1853, 2013.
[47] D. H. Deshmukh, T. Ghorpade, and P. Padiya, "Improving classification using preprocessing and machine learning algorithms on NSL-KDD dataset," in *Proc. Int. Conf. Commun., Inf. Comput. Technol. (ICCICT)*, Jan. 2015, pp. 1–6.
[48] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Jul. 2009, pp. 1–6.
[49] V. Engen, J. Vincent, and K. Phalp, "Exploring discrepancies in findings obtained with the KDD Cup '99 data set,'" *Intell. Data Anal.*, vol. 15, no. 2, pp. 251–276, Mar. 2011.
[50] L. Dhanabal and S. P. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," vol. 4, no. 6, pp. 446–452, 2015.
[51] E. M. Stock, J. D. Stamey, R. Sankaranarayanan, D. M. Young, R. Muwonge, and M. Arbyn, "Estimation of disease prevalence, true positive rate, and false positive rate of two screening tests when disease verification is applied on only screen-positives: A hierarchical model using multi-center data," *Cancer Epidemiol.*, vol. 36, no. 2, pp. 153–160, Apr. 2012.
[52] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.
[53] T. A. Tang, L. Mhamdi, D. Mclernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for network intrusion detection in software defined networking," in *Proc. Int. Conf. Wireless Netw. Mobile Commun. (WINCOM)*, Oct. 2016.
[54] 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.

TONGTONG SU was born in 1992. He received the master's degree in computer science from the School of Computer and Information Engineering, Tianjin Normal University, in 2019. His main research interests include machine learning and pattern recognition.

HUAZHI SUN received the Ph.D. degree from the University of Science and Technology of Beijing, China, in 2008. He is currently a Professor with the School of Computer and Information Engineering, Tianjin Normal University, China. His main research interests include mobile computing and distributed computing.

JINQI ZHU received the Ph.D. degree in computer science from the University of Electronic Science and Technology of China (UESTC), China, in 2009. In 2013, she joined Nanyang Technological University (NTU) as a Visiting Scholar, under the supervision of Dr. Y. G. Wen. She is currently an Associate Professor with the School of Computer and Information Engineering, Tianjin Normal University, China. Her main research interests include mobile computing, vehicular networks, and security networks.

SHENG WANG is currently pursuing the master's degree with the Academy of Computer and Information Engineering, Tianjin Normal University, China. His current research interests include network technology, big data analysis, and deep learning.

YABO LI is currently pursuing the master's degree in computer application technology with Tianjin Normal University. Her main research interests include wireless self-organizing networks and mobile computing.