IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 5, 2017, accepted October 5, 2017, date of publication October 12, 2017, date of current version November 7, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2762418

A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks

CHUANLONG YIN, YUEFEI ZHU, JINLONG FEI, AND XINZHENG HE
State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China
Corresponding author: Chuanlong Yin (dragonyincl@163.com)

This work was supported by the National Key Research and Development Program of China under Grant 2016YFB0801601 and 2016YFB0801505.

ABSTRACT Intrusion detection plays an important role in ensuring information security, and the key technology is to accurately identify various attacks in the network. In this paper, we explore how to model an intrusion detection system based on deep learning, and we propose a deep learning approach for intrusion detection using recurrent neural networks (RNN-IDS). Moreover, we study the performance of the model in binary classification and multiclass classification, and the number of neurons and different learning rate impacts on the performance of the proposed model. We compare it with those of J48, artificial neural network, random forest, support vector machine, and other machine learning methods proposed by previous researchers on the benchmark data set. The experimental results show that RNN-IDS is very suitable for modeling a classification model with high accuracy and that its performance is superior to that of traditional machine learning classification methods in both binary and multiclass classification. The RNN-IDS model improves the accuracy of the intrusion detection and provides a new research method for intrusion detection.

INDEX TERMS Recurrent neural networks, RNN-IDS, intrusion detection, deep learning, machine learning.

I. INTRODUCTION

With the increasingly deep integration of the Internet and society, the Internet is changing the way in which people live, study and work, but the various security threats that we face are becoming more and more serious. How to identify various network attacks, especially unforeseen attacks, is an unavoidable key technical issue. An Intrusion Detection System (IDS), a significant research achievement in the information security field, can identify an invasion, which could be an ongoing invasion or an intrusion that has already occurred. In fact, intrusion detection is usually equivalent to a classification problem, such as a binary or a multiclass classification problem, i.e., identifying whether network traffic behaviour is normal or anomalous, or a five-category classification problem, i.e., identifying whether it is normal or any one of the other four attack types: Denial of Service (DOS), User to Root (U2R), Probe (Probing) and Root to Local (R2L). In short, the main motivation of intrusion detection is to improve the accuracy of classifiers in effectively identifying the intrusive behaviour.

Machine learning methodologies have been widely used in identifying various types of attacks, and a machine learning approach can help the network administrator take the corresponding measures for preventing intrusions. However, most of the traditional machine learning methodologies belong to shallow learning and often emphasize feature engineering and selection; they cannot effectively solve the massive intrusion data classification problem that arises in the face of a real network application environment. With the dynamic growth of data sets, multiple classification tasks will lead to decreased accuracy. In addition, shallow learning is unsuited to intelligent analysis and the forecasting requirements of high-dimensional learning with massive data. In contrast, deep learners have the potential to extract better representations from the data to create much better models. As a result, intrusion detection technology has experienced rapid development after falling into a relatively slow period.

After Professor Hinton [1] proposed the theory of deep learning in 2006, deep learning theory and technology underwent a meteoric rise in the field of machine learning. In this scenario, relevant theoretical papers and practical research findings emerged endlessly and produced remarkable achievements, especially in the fields of speech recognition, image recognition [2] and action recognition [3]–[5]. The fact that deep learning theory and technology has had a very rapid development in recent years means that a new

era of artificial intelligence has opened and offered a completely new way to develop intelligent intrusion detection technology.

Due to growing computational resources, recurrent neural networks (RNNs) (which have been around for decades but their full potential has only recently started to become widely recognized, such as convolutional neural networks (CNNs)) have recently generated a significant development in the domain of deep learning [6]. In recent years, RNNs have played an important role in the fields of computer vision, natural language processing (NLP), semantic understanding, speech recognition, language modelling, translation, picture description, and human action recognition [7]–[9], among others.

Because deep learning has the potential to extract better representations from the data to create much better models, and inspired by recurrent neural networks, we have proposed a deep learning approach for an intrusion detection system using recurrent neural networks (RNN-IDS). The main contributions of this paper are summarized as follows.

(1) We present the design and implementation of the detection system based on recurrent neural networks. Moreover, we study the performance of the model in binary classification and multiclass classification, and the number of neurons and different learning rate impacts on the accuracy.

(2) By contrast, we study the performance of the naive bayesian, random forest, multi-layer perceptron, support vector machine and other machine learning methods in multiclass classification on the benchmark NSL-KDD dataset.

(3) We compare the performance of RNN-IDS with other machine learning methods both in binary classification and multiclass classification. The experimental results illustrate that RNN-IDS is very suitable for intrusion detection. The performance of RNN-IDS is superior to the traditional classification method on the NSL-KDD dataset in both binary and multiclass classification, and it improves the accuracy of intrusion detection, thus providing a new research method for intrusion detection.

The remainder of this paper is organized as follows. In Section II, we review the related research in the field of intrusion detection, especially how deep learning methods facilitate the development of intrusion detection. A description of a RNN-based IDS architecture and the performance evaluation measures are introduced in Section III. Section IV highlights RNN-IDS with a discussion about the experimental results and a comparison with a few previous studies using the NSL-KDD dataset. Finally, the conclusions are discussed in Section V.

II. RELEVANT WORK

In prior studies, a number of approaches based on traditional machine learning, including SVM [10], [11], K-Nearest Neighbour (KNN) [12], ANN [13], Random Forest (RF) [14], [15] and others [16], [17], have been proposed and have achieved success for an intrusion detection system.

In recent years, deep learning, a branch of machine learning, has become increasingly popular and has been applied for intrusion detection; studies have shown that deep learning completely surpasses traditional methods. In [18], the authors utilize a deep learning approach based on a deep neural network for flow-based anomaly detection, and the experimental results show that deep learning can be applied for anomaly detection in software defined networks. In [19], the authors propose a deep learning based approach using self-taught learning (STL) on the benchmark NSL-KDD dataset in a network intrusion detection system. When comparing its performance with those observed in previous studies, the method is shown to be more effective. However, this category of references focuses on the feature reduction ability of the deep learning. It mainly uses deep learning methods for pretraining, and it performs classification through the traditional supervision model. It is not common to apply the deep learning method to perform classification directly, and there is a lack of study of the performance in multiclass classification.

According to [20], RNNs are considered reduced-size neural networks. In that paper, the author proposes a three-layer RNN architecture with 41 features as inputs and four intrusion categories as outputs, and for misuse-based IDS. However, the nodes of layers are partially connected, the reduced RNNs do not show the ability of deep learning to model high-dimensional features, and the authors do not study the performance of the model in the binary classification.

With the continuous development of big data and computing power, deep learning methods have blossomed rapidly, and have been widely utilized in various fields. Following this line of thinking, a deep learning approach for intrusion detection using recurrent neural networks (RNN-IDS) is proposed in this paper. Compared with previous works, we use the RNN-based model for classification rather than for pretraining. Besides, we use the NSL-KDD dataset with a separate training and testing set to evaluate their performances in detecting network intrusions in both binary and multiclass classification, and we compare it with J48, ANN, RF, SVM and other machine learning methods proposed by previous researchers.

III. PROPOSED METHODOLOGIES

Recurrent neural networks include input units, output units and hidden units, and the hidden unit completes the most important work. The RNN model essentially has a one-way flow of information from the input units to the hidden units, and the synthesis of the one-way information flow from the previous temporal concealment unit to the current timing hiding unit is shown in Fig. 1. We can regard hidden units as the storage of the whole network, which remember the end-to-end information. When we unfold the RNN, we can find that it embodies the deep learning. A RNNs approach can be used for supervised classification learning.

Recurrent neural networks have introduced a directional loop that can memorize the previous information and apply

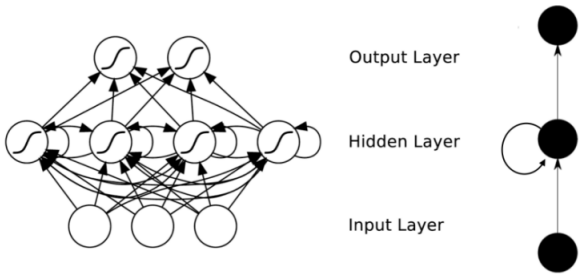


FIGURE 1. Recurrent Neural Networks (RNNs).

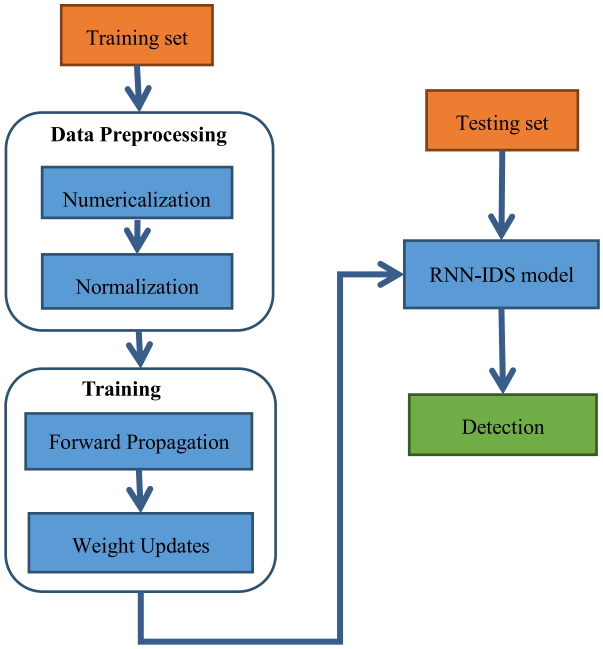


FIGURE 2. Block diagram of proposed RNN-IDS.

it to the current output, which is the essential difference from traditional Feed-forward Neural Networks (FNNs). The preceding output is also related to the current output of a sequence, and the nodes between the hidden layers are no longer connectionless; instead, they have connections. Not only the output of the input layer but also the output of the last hidden layer acts on the input of the hidden layer.

The step involved in RNN-IDS is shown in Fig. 2.

A. DATASET DESCRIPTION

The NSL-KDD dataset [21], [22] generated in 2009 is widely used in intrusion detection experiments. In the latest literature [23]–[25], all the researchers use the NSL-KDD as the benchmark dataset, which 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 set and testing set, in such a way that the classifier does not favour more frequent records. The dataset covers the KDDTrain^{+} dataset as the training set and KDDTest^{+} and KDDTest^{-21} datasets as the testing set, which has different

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

	Total	Normal	Dos	Probe	R2L	U2R
KDDTrain^{+}	125973	67343	45927	11656	995	52
KDDTest^{+}	22544	9711	7458	2421	2754	200
KDDTest^{-21}	11850	2152	4342	2402	2754	200

TABLE 2. Features of NSL-KDD dataset.

No.	Features	Types	No.	Features	Types
1	duration	Continuous	22	is_guest_login	Symbolic
2	protocol_type	Symbolic	23	count	Continuous
3	service	Symbolic	24	srv_count	Continuous
4	flag	Symbolic	25	serror_rate	Continuous
5	src_bytes	Continuous	26	srv_serror_rate	Continuous
6	dst_bytes	Continuous	27	rerror_rate	Continuous
7	land	Symbolic	28	srv_rerror_rate	Continuous
8	wrong_fragment	Continuous	29	same_srv_rate	Continuous
9	urgent	Continuous	30	diff_srv_rate	Continuous
10	hot	Continuous	31	srv_diff_host_rate	Continuous
11	num_failed_logins	Continuous	32	dst_host_count	Continuous
12	logged_in	Symbolic	33	dst_host_srv_count	Continuous
13	num_compromised	Continuous	34	dst_host_same_srv_rate	Continuous
14	root_shell	Continuous	35	dst_host_diff_srv_rate	Continuous
15	su_attempted	Continuous	36	dst_host_same_src_port_ra	Continuous
16	num_root	Continuous	37	dst_host_srv_diff_host_rat	Continuous
17	num_file_creations	Continuous	38	dst_host_serror_rate	Continuous
18	num_shells	Continuous	39	dst_host_srv_serror_rate	Continuous
19	num_access_files	Continuous	40	dst_host_rerror_rate	Continuous
20	num_outbound_cmds	Continuous	41	dst_host_srv_rerror_rate	Continuous
21	is_host_login	Symbolic			

normal records and four different types of attack records, as shown in Table 1. The KDDTest^{-21} dataset is a subset of the KDDTest^{+} and is more difficult for classification.

There are 41 features and 1 class label for every traffic record, and the features include basic features (No.1-No.10), content features (No.11 - No.22), and traffic features (No.23 - No.41) as shown in Table 2. According to their characteristics, attacks in the dataset are categorized into four attack types: DoS (Denial of Service attacks), R2L (Root to Local attacks), U2R (User to Root attack), and Probe (Probing attacks). The testing set has some specific attack types that disappear in the training set, which allows it to provide a more realistic theoretical basis for intrusion detection.

B. DATA PREPROCESSING

1) NUMERICALIZATION

There are 38 numeric features and 3 nonnumeric features in the NSL-KDD dataset. Because the input value of RNN-IDS should be a numeric matrix, we must convert some nonnumeric features, such as 'protocol_type', 'service' and 'flag' features, into numeric form. For example, the feature 'protocol_type' has three types of attributes, 'tcp', 'udp', and 'icmp', and its numeric values are encoded as binary

vectors (1,0,0), (0,1,0) and (0,0,1). Similarly, the feature 'service' has 70 types of attributes, and the feature 'flag' has 11 types of attributes. Continuing in this way, 41-dimensional features map into 122-dimensional features after transformation.

2) NORMALIZATION

First, according to some features, suchas 'duration[0,58329]', 'src_bytes[0,1.3 × 109]' and 'dst_bytes[0,1.3 × 109]', where the difference between the maximum and minimum values has a very large scope, we apply the logarithmic scaling method for scaling to obtain the ranges of 'duration[0,4.77]', 'src_bytes[0,9.11]' and 'dst_bytes[0,9.11]'. Second, the value of every feature is mapped to the [0,1] range linearly according to (1), where Max denotes the maximum value and Min denotes minimum value for each feature.

$$x_i = \frac{x_i - Min}{Max - Min} \tag{1}$$

C. METHODOLOGY

It is obvious that the training of the RNN-IDS model consists of two parts - Forward Propagation and Back Propagation. Forward Propagation is responsible for calculating the output values, and Back Propagation is responsible for passing the residuals that were accumulated to update the weights, which is not fundamentally different from the normal neural network training.

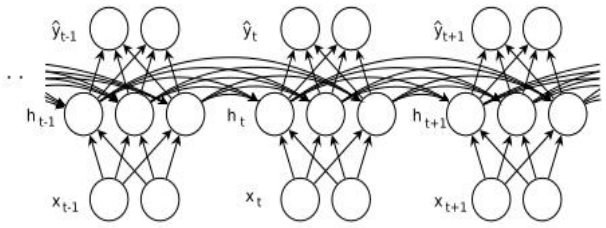


FIGURE 3. The unfolded Recurrent Neural Network.

According to Fig. 1, an unfolded recurrent neural network is presented in Fig. 3. The standard RNN is formalized as follows: Given training samples $x_i(i = 1, 2, \ldots, m)$, a sequence of hidden states h_i $(i = 1, 2, \ldots, m)$, and a sequence of predictions $\hat{y}_i(i = 1, 2, \ldots, m)$. $\mathbf{W}_{hx}$ is the input-to-hidden weight matrix, $\mathbf{W}_{hh}$ is the hidden-to-hidden weight matrix, $\mathbf{W}_{yh}$ is the hidden-to-output weight matrix, and the vectors b_h and b_y are the biases [26]. The activation function $\boldsymbol{e}$ is a *sigmoid*, and the classification function $\boldsymbol{g}$ engages the *SoftMax* function.

Refer to Fig. 3 and [26], Forward Propagation Algorithm and Weights Update Algorithm are described as Algorithms 1 and 2 respectively.

The objective function associated with RNNs for a single training pair (x_i, y_i) is defined as $f(\theta) = \mathbf{L}(y_i : \hat{y}_i)$ [26], where $\mathbf{L}$ is a distance function which measures the deviation of the predictions $\hat{y}_i$ from the actual labels y_i. Let η be the learning rate and k be the number of current iterations. Given a sequence of labels $y_i(i = 1, 2, \ldots, m)$.

Algorithm 1 Forward Propagation Algorithm

Input $x_i(i = 1, 2, \ldots, m)$
Output $\hat{y}_i$
1: for i from *1* to m do
2: $t\iota = \mathbf{W}_{hxxi} + \mathbf{W}_{hhhi-1+bh}$
3: $hi = \text{sigmoid}\ (ti)$
4: $si = \mathbf{W}_{yhhi+by}$
5: $\hat{y}i = \text{SoftMax}\ (si)$
6: end for

Algorithm 2 Weights Update Algorithm

Input $\langle y_i, \hat{y}_i\rangle(i = 1, 2, \ldots, m)$
Initialization $\theta = \{\mathbf{W}_{hx}, \mathbf{W}_{hh}, \mathbf{W}_{yh}, b_h, b_y\}$
Output $\theta = \{\mathbf{W}_{hx}, \mathbf{W}_{hh}, \mathbf{W}_{yh}, b_h, b_y\}$
1: for i from k downto *1* do
2: Calculate the cross entropy between the output value and the label value: $L(y_i: \hat{y}_i) \leftarrow -\sum_i \sum_j y_{ij} \log\ (\hat{y}_{ij}) + (1 - y_{ij}) \log(1 - \hat{y}_{ij})$
3: Compute the partial derivative with respect to θ_i : $\delta_i \leftarrow dL/d\theta_i$
4: Weight update: $\theta_i \leftarrow \theta_i\eta + \delta_i$
5: end for

D. EVALUATION METRICS

In our model, the most important performance indicator (Accuracy, AC) of intrusion detection is used to measure the performance of the RNN-IDS model. In addition to the accuracy, we introduce the detection rate and false positive rate. The True Positive (TP) is equivalent to those correctly rejected, and it denotes the number of anomaly records that are identified as anomaly. The False Positive (FP) is the equivalent of incorrectly rejected, and it denotes the number of normal records that are identified as anomaly. The True Negative (TN) is equivalent to those correctly admitted, and it denotes the number of normal records that are identified as normal. The False Negative (FN) is equivalent to those incorrectly admitted, and it denotes the number of anomaly records that are identified as normal. Table 3 shows the definition of confusion matrix. We have the following notation:

Accuracy: the percentage of the number of records classified correctly versus total the records shown in (2).

$$\text{AC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2}$$

True Positive Rate (TPR): as the equivalent of the Detection Rate (DR), it shows the percentage of the number of records identified correctly over the total number of anomaly records, as shown in (3).

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

False Positive Rate (FPR): the percentage of the number of records rejected incorrectly is divided by the total number of

TABLE 3. Confusion matrix.

Actual Class \ Predicted Class	anomaly	normal
anomaly	TP	FN
normal	FP	TN

normal records, as shown in (4).

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{4}$$

Hence, the motivation for the IDS is to obtain a higher accuracy and detection rate with a lower false positive rate.

IV. EXPERIMENT RESULTS AND DISCUSSION

In this research, we have used one of the most current and broadest deep learning frameworks - Theano [27]. The experiment is performed on a personal notebook ThinkPad E450, which has a configuration of an Intel Core i5-5200U CPU @ 2.20 GHz, 8 GB memory and does not use GPU acceleration. Two experiments have been designed to study the performance of the RNN-IDS model for binary classification (Normal, anomaly) and five-category classification, such as Normal, DoS, R2L, U2R and Probe. In order to compare with other other machine learning methods, contrast experiments are designed at the same time. In the binary classification experiments, we have compared the performance with an ANN, naive Bayesian, random forest, multi-layer perceptron, support vector machine and other machine learning methods, as mentioned in [13] and [21]. In the same way, we analyse the multi-classification of the RNN-IDS model based on the NSL-KDD dataset. By contrast, we study the performance of the ANN, naive Bayesian, random forest, multi-layer perceptron, support vector machine and other machine learning methods in the five-category classification. Finally, we compare the performance of the RNN-IDS model with traditional methods. Furthermore, we construct the dataset refer to [20] and compare the performance with the reduced-size RNN method.

A. BINARY CLASSIFICATION

In Sec B, we have mapped 41-dimensional features into 122-dimensional features, thus the RNN-IDS model has 122 input nodes, and 2 output nodes in the binary classification experiments. The number of epochs are given 100. To train the better model, let the number of hidden nodes be 20, 60, 80, 120, and 240 respectively, the learning rate be 0.01, 0.1 and 0.5 respectively, then we observe the classification accuracy on the NSL-KDD dataset as shown in Table 4. The different results we obtain show that the accuracy is relate to the number of hidden nodes and the learning rate.

In our experiment, the model gets a higher accuracy, when there are 80 hidden nodes and the learning rate is 0.1. Table 5 shows the confusion matrix of the RNN-IDS on the testing set KDDTest$^+$ in the 2-category classification experiments. The experiments show that RNN-IDS works with a good detection rate (83.28%) when given 100 epochs for the KDDTrain$^+$ dataset. We obtain 68.55% for the KDDTest^{-21} dataset and 99.81% for the KDDTrain$^+$ dataset as shown in Fig. 4.

TABLE 4. The accuracy and training time (second) of RNN-IDS with different learning rate and hidden nodes.

	KDDTrain$^+$	KDDTest$^+$	KDDTest^{-21}	Time
Hidden Nodes = 20, learning rate =0.01	99.40%	79.37%	60.76%	4155
Hidden Nodes = 20, learning rate =0.1	99.79%	83.18%	68.23%	3900
Hidden Nodes = 20, learning rate =0.5	99.81%	83.09%	67.84%	3331
Hidden Nodes = 60, learning rate =0.01	99.39%	78.72%	59.54%	4135
Hidden Nodes = 60, learning rate =0.1	99.79%	81.06%	64.08%	4613
Hidden Nodes = 60, learning rate =0.5	99.87%	83.11%	67.82%	3946
Hidden Nodes = 80, learning rate =0.01	99.29%	79.16%	60.34%	4324
Hidden Nodes = 80, learning rate =0.1	99.81%	83.28%	68.55%	5516
Hidden Nodes = 80, learning rate =0.5	99.85%	82.66%	66.99%	4478
Hidden Nodes = 120, learning rate =0.01	99.28%	78.55%	59.25%	4786
Hidden Nodes = 120, learning rate =0.1	99.79%	82.48%	66.83%	5868
Hidden Nodes = 120, learning rate =0.5	99.87%	80.97%	63.69%	5107
Hidden Nodes = 240, learning rate =0.01	99.69%	80.69%	63.28%	12203
Hidden Nodes = 240, learning rate =0.1	99.69%	80.67%	63.28%	10966
Hidden Nodes = 240, learning rate =0.5	99.87%	80.97%	63.69%	7836

TABLE 5. Confusion matrix of 2-category classification on KDDTEST$^+$.

Actual Class \ Predicted Class	anomaly	normal
anomaly	9362	3471
normal	298	9413

In [21], the authors have shown the results obtained by J48, Naive Bayesian, Random Forest, Multi-layer Perceptron, Support Vector Machine and the other classification algorithms, and the artificial neural network algorithm also gives 81.2% in [13], which is the recent literature about ANN algorithms applied in the filed of intrusion detection. Fortunately, these results are all based on the same benchmark - the NSL-KDD dataset. Obviously, the performance of RNN-IDS model is superior to other classification algorithms in binary classification as shown in Fig. 5.

B. MULTICLASS CLASSIFICATION

In the five-category classification experiments, we find that the model has higher accuracy on the KDDTest$^+$ when there are 80 hidden nodes in the RNN-IDS model, meanwhile the

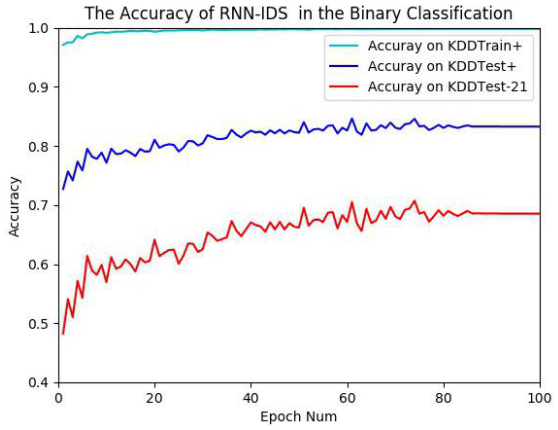


FIGURE 4. The Accuracy on the KDDTest^{+} and KDDTest^{-21} datasets in the Binary Classification.

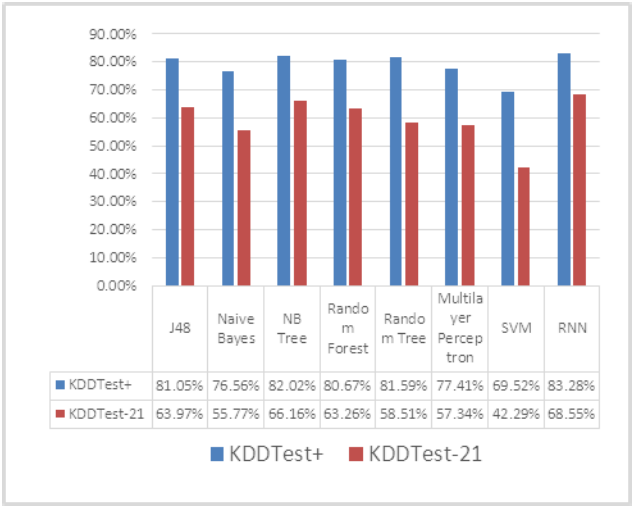


FIGURE 5. Performance of RNN-IDS and the other models in the binary classification.

learning rate is 0.5, and the training is performed 80 times from Table 6.

In order to compare the performance of different classification algorithms on the benchmark dataset for the multi-calss classification as the binary classification experiments J48, Naive Bayesian, Random Forest, Multi-layer Perceptron, Support Vector Machine and other machine learning algorithms are used to train models through the training set (using 10-layer cross-validation) by mean of the open-source machine learning and data mining software Weka [28]. We then apply the models to the testing set. The results are described in Fig. 6. Compared with the binary classification, the accuracy of classification algorithms is declined in the five-category classification.

Table 7 shows the confusion matrix of the RNN-IDS on the test set KDDTest^{+} in the five-category classification experiments. The experiment shows that the accuracy of the model is 81.29% for the test set KDDTest^{+} and 64.67% for KDDTest^{-21}, which is better than those obtained using J48,

TABLE 6. The accuracy and training time (second) of RNN-IDS with different learning rate and hidden nodes.

	KDDTrain^{+}	KDDTest^{+}	KDDTest^{-21}	Time
Hidden Nodes = 60, learning rate =0.1	99.84%	79.87%	61.98%	8065
Hidden Nodes = 60, learning rate =0.5	99.87%	77.46%	57.18%	9855
Hidden Nodes = 60, learning rate =0.8	91.23%	69.29%	41.85%	10771
Hidden Nodes = 80, learning rate =0.1	99.82%	77.73%	57.84%	9680
Hidden Nodes = 80, learning rate =0.5	99.53%	81.29%	64.67%	11444
Hidden Nodes = 80, learning rate =0.8	98.97%	77.09%	56.64%	13622
Hidden Nodes = 120, learning rate =0.1	99.85%	77.02%	56.55%	12218
Hidden Nodes = 120, learning rate =0.5	99.87%	79.44%	61.11%	14404
Hidden Nodes = 120, learning rate =0.8	93.90%	70.32%	45.49%	17534
Hidden Nodes = 160, learning rate =0.1	99.68%	77.85%	58.05%	15534
Hidden Nodes = 160, learning rate =0.5	99.80%	78.73%	59.71%	17540
Hidden Nodes = 160, learning rate =0.8	92.79%	71.15%	45.64%	22901

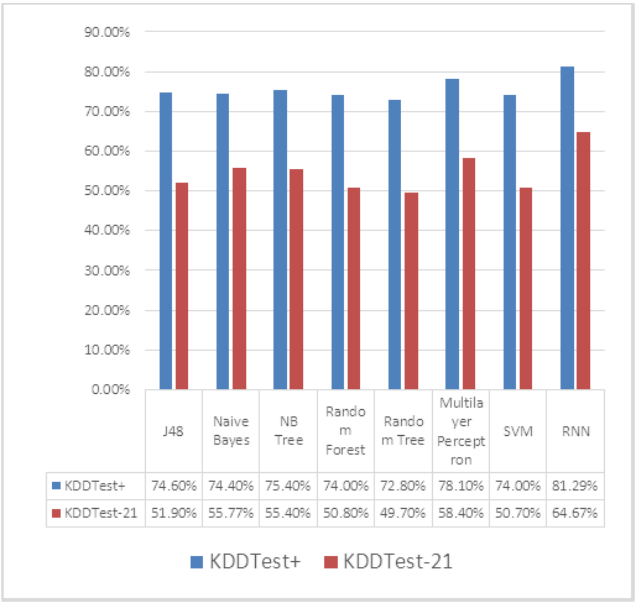


FIGURE 6. Performance of RNN-IDS and the other models in the five-category classification.

naive bayes, random forest, multi-layer perceptron and the other classification algorithms. In addition, it is better than the artificial neural network algorithm on the test set KDDTest^{+}, which obtained 79.9% in the literature [13]. Table 8 shows the detection rate and false positive rate of the different attack types.

In order to compare the performance of RNN-IDS with the reduced-size RNN method proposed in [20], we constructed the training set and testing set from KDD CUP 1999 dataset according to the paper. The training and testing

TABLE 7. Confusion matrix for the five-category experiments on KDDTest$^+$.

Actual Class \ Predicted Class	Normal	DoS	R2L	U2R	Probe
Normal	9377	88	2	6	238
DoS	1011	6227	125	0	95
R2L	2058	0	680	6	10
U2R	149	0	11	23	17
Probe	231	166	5	0	2019

TABLE 8. Results of the evaluation metrics for the five-category classification.

Intrusion Type	FPR(%)	DR(%)
DoS	2.06	83.49
R2L	0.80	24.69
U2R	0.07	11.50
Probe	2.16	83.40

TABLE 9. Different classifications in the training and testing sets

Class	Number of training samples	Number of testing samples
Normal	9,727	6,059
DoS	39,145	22,985
Probe	411	417
U2R	6	24
R2L	113	1,619

sets are described in detail in Table 9. In this experiment, the detection rate of the RNN-IDS model gets 97.09% on the testing dataset, not only higher than the detection rate on the NSL-KDD dataset, but also higher than 94.1% in the literature [20]. The experimental results show that the fully connected model has stronger modeling ability and higher detection rate than the reduced-size RNN model. The training of our model (20 hidden nodes, the learning rate is 0.1, and epochs are 50) spends 1765 seconds without any GPU acceleration, which more than 1383 seconds in the literature [20].

C. DISCUSSION

Based on the same benchmark, using KDDTrain$^+$ as the training set and KDDTest$^+$ and KDDTest^{-21} as the testing set, the experimental results show that for both binary and multiple classification, the intrusion detection model of RNN-IDS training through the training set has higher accuracy than the other machine learning methods and maintains a high accuracy rate, even in the case of multiple classification. Of course, the model we proposed will spend more time for training, but using GPU acceleration can reduce the training time.

V. CONCLUSIONS

The RNN-IDS model not only has a strong modelling ability for intrusion detection, but also has high accuracy in both binary and multiclass classification. 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. The model can effectively improve both the accuracy of intrusion detection and the ability to recognize the intrusion type. Of course, in the future research, we will still pay attention to reduce the training time using GPU acceleration, avoid exploding and vanishing gradients, and study the classification performance of LSTM, Bidirectional RNNs algorithm in the field of intrusion detection.

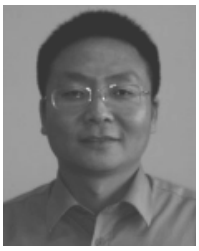
REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
[2] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.
[3] L. Liu, L. Shao, X. Li, and K. Lu, "Learning spatio-temporal representations for action recognition: A genetic programming approach," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 158–170, Jan. 2016.
[4] A.-A. Liu, Y.-T. Su, W.-Z. Nie, and M. Kankanhalli, "Hierarchical clustering multi-task learning for joint human action grouping and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 102–114, Jan. 2017.
[5] J. Wu, Y. Zhang, and W. Lin, "Good practices for learning to recognize actions using FV and VLAD," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2978–2990, Dec. 2016.
[6] A. Karpathy. (2015). The unreasonable effectiveness of recurrent neural networks. Andrej Karpathy Blog. [Online]. Available: http://karpathy.github.io/2015/05/21/rnn-effectiveness/
[7] X. Peng, L. Wang, X. Wang, and Y. Qiao, "Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice," *Comput. Vis. Image Understand.*, vol. 150, pp. 109–125, Sep. 2016.
[8] A.-A. Liu, Y.-T. Su, P.-P. Jia, Z. Gao, T. Hao, and Z.-X. Yang, "Multiple/single-view human action recognition via part-induced multitask structural learning," *IEEE Trans. Cybern.*, vol. 45, no. 6, pp. 1194–1208, Jun. 2015.
[9] W. Nie, A. Liu, W. Li, and Y. Su, "Cross-view action recognition by cross-domain learning," *Image Vis. Comput.*, vol. 55, pp. 109–118, Nov. 2016.
[10] 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.
[11] R. R. Reddy, Y. Ramadevi, and K. V. N. Sunitha, "Effective discriminant function for intrusion detection using SVM," in *Proc. Int. Conf. Adv. Comput., Commun. Inform. (ICACCI)*, Sep. 2016, pp. 1148–1153.
[12] 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. Elect. Comput. Eng.*, vol. 2014, Jun. 2014, Art. no. 240217.
[13] B. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN," in *Proc. Int. Conf. Signal Process. Commun. Eng. Syst.*, Jan. 2015, pp. 92–96.
[14] N. Farnaaz and M. A. Jabbar, "Random forest modeling for network intrusion detection system," *Procedia Comput. Sci.*, vol. 89, pp. 213–217, Jan. 2016.
[15] 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.
[16] J. A. Khan and N. Jain, "A survey on intrusion detection systems and classification techniques," *Int. J. Sci. Res. Sci., Eng. Technol.*, vol. 2, no. 5, pp. 202–208, 2016.

[17] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2nd Quart., 2016.
[18] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," presented at the 9th EAI Int. Conf. Bio-inspired Inf. Commun. Technol. (BIONETICS), New York, NY, USA, May 2016, pp. 21–26.
[19] 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, pp. 258–263.
[20] M. Sheikhan, Z. Jadidi, and A. Farrokhi, "Intrusion detection using reduced-size RNN based on feature grouping," *Neural Comput. Appl.*, vol. 21, no. 6, pp. 1185–1190, Sep. 2012.
[21] M. Tavallaee, E. Bagheri, W. Lu, and A. 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.
[22] 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, pp. 1848–1853, Dec. 2013.
[23] N. Paulauskas and J. Auskalnis, "Analysis of data pre-processing influence on intrusion detection using NSL-KDD dataset," in *Proc. Open Conf. Elect., Electron. Inf. Sci. (eStream)*, Apr. 2017, pp. 1–5.
[24] P. S. Bhattacharjee, A. K. M. Fujail, and S. A. Begum, "Intrusion detection system for NSL-KDD data set using vectorised fitness function in genetic algorithm," *Adv. Comput. Sci. Technol.*, vol. 10, no. 2, pp. 235–246, 2017.
[25] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Inf. Sci.*, vol. 378, pp. 484–497, Feb. 2017.
[26] J. Martens and I. Sutskever, "Learning recurrent neural networks with hessian-free optimization," presented at the 28th Int. Conf. Int. Conf. Mach. Learn., Bellevue, WA, USA, Jul. 2011, pp. 1033–1040.
[27] *Welcome: Theano 0.9.0 Documentation*. Accessed: Feb. 2017. [Online]. Available: http://deeplearning.net/software/theano/
[28] *Weka 3–Data Mining With Open Source Machine Learning Software in Java*. Accessed: Dec. 2016. [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/

CHUANLONG YIN was born in 1985. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Mathematical Engineering and Advanced Computing. His research areas are intrusion detection and information security.

YUEFEI ZHU was born in 1962. He is currently a Professor and a Doctoral Supervisor with the State Key Laboratory of Mathematical Engineering and Advanced Computing. His research areas are intrusion detection, cryptography, and information security.



JINLONG FEI was born in 1980. He is currently an Associate Professor with the State Key Laboratory of Mathematical Engineering and Advanced Computing. His research areas are network traffic analysis and information security.

XINZHENG HE was born in 1978. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Mathematical Engineering and Advanced Computing. His research areas are big data and information security.