Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 1419804, 10 pages
https://doi.org/10.1155/2022/1419804

Research Article

Identification of Encrypted Traffic Using Advanced Mathematical Modeling and Computational Intelligence

Xinlei Liu

School of Network and Information Security, Xidian University, Xi'an 710126, China

Correspondence should be addressed to Xinlei Liu; 19180100102@stu.xidian.edu.cn

Received 15 June 2022; Accepted 12 July 2022; Published 22 August 2022

Academic Editor: Dragan Pamučar

This paper proposed a hybrid approach for the identification of encrypted traffic based on advanced mathematical modeling and computational intelligence. Network traffic identification is the premise and foundation of improving network management, service quality, and application security. It is also the focus of network behavior analysis, network planning and construction, network anomaly detection, and network traffic model research. With the increase in user and service requirements, many applications use encryption algorithms to encrypt traffic during data transmission. As a result, traditional traffic classification methods classify encrypted traffic on the network, which brings great difficulties and challenges to network monitoring and data mining. In our article, a nonlinear modified DBN method is proposed and applied to encrypted traffic identification. Firstly, based on Deep Belief Networks (DBN), this paper introduces the proposed Eodified Elliott (ME)-DBN model, analyzes the function image, and presents the ME-DBN learning algorithm. Secondly, this article designs an encrypted traffic recognition model based on the ME-DBN model. Feature extraction is carried out by training the ME-DBN model, and finally, classification and recognition are carried out by the classifier. The experimental results on the ISCX VPN-non-VPN database show that the MEDBN method proposed in this article can enhance the classification and recognition rate and has better robustness to encrypt traffic recognition from different software.

1. Introduction

Network traffic classification is a basic step for managing and controlling network resources. Previous traffic classification methods, such as the traffic classification method [1] based on port number and deep Packet Inspection (DPI), cannot deal with encrypted traffic and can hardly adapt to the current traffic environment. The method based on traffic statistics and machine learning (ML) is popular in current, which not only can deal with encrypted traffic but also regular traffic, for example, decision tree (DT) and KNN algorithm. Nevertheless, the performance based on the ML depends largely on artificially designed features and private information in traffic. Therefore, there is a limitation on the generality and accuracy of the method. In addition, the method *t* requires a mass of storage and computing resources, which limits its implementation in resource-constrained nodes [2], such as vehicles, home gateways, and mobile phones. Real-time and accurate network traffic classification is the basis of network management tasks and intrusion detection systems, so a new traffic identification method is urgently needed.

The development of mobile traffic identification technology has experienced three stages based on port, based on payload and based on traffic statistical characteristics. Nevertheless, the advent of port spoofing, random ports, and tunneling quickly rendered these models ineffective. As users become more aware of privacy protection and security, technologies such as SSL, SSH, VPN, and Tor have become more widely used, resulting in an increasing proportion of encrypted traffic in network traffic. The payload-based approach, known as Deep Packet Inspection (DPI) technology, cannot handle encrypted traffic because it requires matching Packet content and is computationally expensive. Therefore, to handle the problem of encrypted traffic classification, the method based on data traffic appears. Its generality depends on statistical or time series properties and uses ML algorithms, like some tree-based methods, classical model as SVM, and KNN, etc. Furthermore, some statistical methods such as GMM [3] and HMM [4] are also employed to identify the encrypted traffic. Classical machine learning methods could handle some problems that port and payload-based methods cannot solve, but they still have certain limitations: (1) The characteristics of data traffic need to be extracted manually, which often depends on expert experience and is very time-consuming and labor-intensive. (2) The characteristics of traffic change rapidly and need to be updated frequently. (3) For traffic identification tasks, category imbalance is a major problem. Category imbalance refers to the fact that the data volume of some samples in the data set is several times or even higher than that of others. Using such data set to train the model, a high recognition rate can be obtained as long as all the small samples are classified into large samples, which is not meaningful in actual production. The method to solve this problem is to expand the data amount of small samples through different ways, but the current data expansion method cannot accurately generate samples as close as possible to the original data. (4) In model training, marked samples are mostly relied on. How to combine a large number of easily obtained unlabeled data sets with some difficult-to-obtain labeled data sets for traffic classification in order to reduce the need for labeled data is a very key research topic. Different from most traditional ML algorithms, DL automatically extracts features without human intervention, which is undoubtedly an ideal traffic classification method, especially for mobile service encrypted traffic. Recent research work proves the superiority of the DL method in traffic classification [5–9]. Therefore, it is of great importance and far-reaching significance to study the application of DL in traffic classification and how to improve the recognition rate of small sample traffic in unbalance data sets so as to more effectively and conveniently encrypt traffic and improve the accuracy of application identification.

Recently, DL-based methods have been employed in many fields and achieved good results, such as image recognition, speech recognition, and natural language processing. Owning to the Deep Learning, this article proposes a frame of classification and detection based on Deep Learning (DL), which can construct feature space through the deep structure of multiple hidden layers and discover data features through autonomous learning of a large number of data. It solves the difficulty of feature subset selection and improves the classification efficiency, which lays a foundation for the real-time classification of network traffic. In the second part, we summarize the existing research. In the third part, we introduce the identification model of encrypted traffic; in the fourth part, we introduce the evaluation process of the model in detail; the fifth part describes the data collection and processing in detail; the sixth part introduces the detailed process of experiment and simulation; the summary and discussion are arranged in the last part.

2. Research Overview

As early as 1995, Claffy et al. [10] used the traditional classification method based on service host attributes to identify network traffic. Almost all communication protocol packets, including encrypted packets, have their own unique traffic characteristics, which can be analyzed and distinguished from a large number of traffic samples. Therefore, Gu et al. [11] made an in-depth analysis of the classification method of traffic load content characteristics. Yeganeh et al. [12] summarized the relative positional word set carried in the session flow payloads of each protocol and then detected the payloads by the deep packet detection method of word sequence matching to identify the protocol types as smart computing continues to evolve.

Due to the quantitative limitations of current technologies, new methods have been found that rely on the statistical characteristics of traffic to classify applications. In recent years, stream classification methods based on statistical features have attracted extensive attention. Common statistical features such as packet interval statistics, flow arrival time statistics, flow duration, packet length, traffic idle time, packet arrival interval, packet length, and other statistical characteristics of the network. With the explosive growth of traffic in the current network environment, the simple traffic statistics method has been unable to achieve the ideal classification effect of network traffic, and the method based on machine learning came into being. Machine learning mainly includes supervised learning, semi-supervised learning, and unsupervised learning.

Recognition models based on ML and DL are widely used. Ibrahim et al. [13] designed a classifier for online traffic classification (SSPC) that combines three identification methods: port-based, payload-based, and statistically based. The classification results based on payload are preferred for identification, followed by the same results based on port and statistical characteristics. Conti et al. [14] used the method of RF to identify the actions of users on mobile phones through the IP, packet size, port, direction and other characteristics of the encrypted traffic generated by marked users when using the application mobile phone client. Compared with the ML-based methods, in 2004, literature [15] used packet length, packet interval, and stream duration as statistical features and used an expectation maximization algorithm to classify traffic types by unsupervised learning. Literature [16] uses an unsupervised machine learning algorithm to carry out unsupervised machine learning training on long-term and short-term memory recurrent neural networks so that the network can distinguish a group of time series and group them. The results show that the neural network has a strong time series learning ability and clustering ability based on multiple features. Literature [17] proposed a method of malicious traffic detection using representation learning. This method does not need to manually design traffic characteristics but directly classifies the original data as input data. This is the first time that the representation learning method is applied to the detection and classification of malicious traffic. When the three classifiers are verified in two cases, the results meet the requirements of practical application accuracy. This document proves that the efficiency of representation learning in malicious traffic detection is high, but there are also shortcomings. The tuning parameters of the convolutional neural network are not studied, and the time factor and unknown malicious traffic are not considered [18]. Literature [19] classifies more than 20 kinds of fine-grained network traffic based on hierarchical learning. The results of large data sets show that the average accuracy of traffic classification of hierarchical classifier can reach 90%, and the accuracy and recall of commonly used traffic categories are higher than 95%. Wang et al. used the long-term and short-term memory recurrent neural network to automatically learn the timing characteristics in the traffic, solved the problem of manually designing the characteristics, and achieved a high detection rate and low false alarm rate. In the literature, a cyclic neural network is used to learn the timing characteristics of encrypted traffic to realize the mobile application type recognition of the Android platform, and a high recognition rate and recall rate are achieved. Literature [5] uses the improved RNN and density clustering method to detect network abnormal traffic, which has achieved better results than the current method. Document [20] introduces a deep packet, which is an algorithm that uses deep learning [21] to automatically extract features from network traffic to classify traffic. The deep packet is the first traffic classification system using a deep learning algorithm, namely SAE and a one-dimensional convolutional neural network, which can identify applications and handle traffic characterization tasks. The automatic feature extraction process of network traffic can save the cost of using experts to identify traffic and extract manual features and reduce the overhead of traffic classification. The classification method based on machine learning has high classification accuracy and can be used for the identification of encrypted traffic, but the cost is high, and the data set needs to be understood and preprocessed in advance. Different business types have different requirements for the packet size of data flow. For example, the flow media data is small, and the packet downloaded from the file can be the maximum message segment length. Therefore, there are differences in the distribution of packet sizes for different business types. The method based on packet size distribution is not affected by encryption and has good applicability. Qin et al. propose a new method based on packet size distribution signature, which can reduce the amount of packet processing and realize the accurate identification of P2P and VolP applications. Renyi cross-entropy is used to identify by calculating the similarity between the two-way flow and the message size distribution of specific applications [22]. Wang et al. [5] simultaneously used CNN and LSTM to learn and classify data packet headers and loads, showing good performance in real-time intrusion detection. Aceto et al. [23, 24] designed and implemented a recognition model based on MLP in order to track which APP the data stream came from. They used some features in the first N bytes of payload and original data, and some features in the first 20 packets before interactive communication, including source port, payload bytes, size of TCP slide window, Sequential packet arrival interval, and direction, which were used as input, and the experiments were compared with random forest, stack automatic encoder SAE, CNN, and LSTM. Martin et al. [25] conducted a group of controlled experiments, respectively, using the combination of RNN, CNN and recursive neural network RNN and CNN to identify the traffic of the Internet of things. The results showed that the combination of RNN and CNN had the best effect. Hochst et al. [26] designed and implemented an autoencoder SAE in order to find out actions such as web browsing interaction, game download, online play, and upload in network traffic, which achieved good results. It can be seen that using deep learning to classify encrypted traffic is a good research direction.

3. Encrypted Traffic Classification Model Design

3.1. Restricted Boltzmann Machine. The constrained Boltzmann machine (RBM) is a deformed structure of the Boltzmann machine (BM). Based on statistical mechanics, the sample of BM follows the Boltzmann machine distribution. The probability distribution of the energy-based probability model is defined by the energy function $E(x)$:

$$P(x) = \frac{e^{-E(x)}}{Z}, \tag{1}$$

where x is the input sample, $Z = \sum_x e^{-E(x)}$ is the normalized function, the commonly used method to solve $P(x)$ is gradient descent, and the negative logarithm of the training set D is its cost function:

$$l(\theta, D) = -L(\theta, D) = -\frac{1}{N} \sum_{x^{(i)} \in D} \log, \tag{2}$$

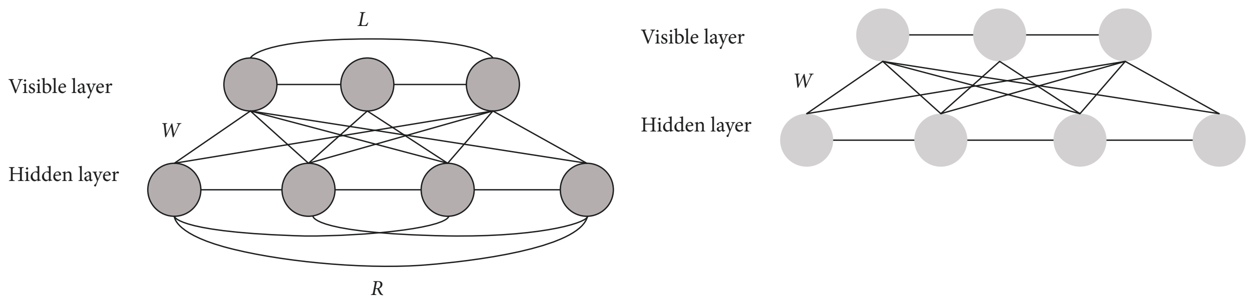
where θ is the parameter space of the model, the partial derivative of θ is obtained through the optimization algorithm so as to get the optimal solution of the cost function:

$$\Delta = \frac{\partial l(\theta, D)}{\partial \theta} = -\frac{1}{N} \frac{\partial \sum \log\, p\left(x^{(i)}\right)}{\partial \theta}. \tag{3}$$

The boltzmann machine is a random NN defined by the above energy function, which consists of a visible layer and a hidden layer, as introduced in Figure 1(a). As can be seen from the figure, both intralayer nodes and interlayer nodes have connection weights, and there are only two states of the output node: activated and inactive. 1 means activated, and 0 means inactive. Therefore, we can see unit vector $v = \{0, 1\}^D$ and implicit unit vector $h = \{0, 1\}^k$, and their learning mode belongs to unsupervised learning. The energy function between the visible layer neuron and hidden layer neuron of the BM model is defined as follows:

$$E(v, h; \theta) = -v^T W h - \frac{1}{2} v^T L v - \frac{1}{2} h^T L h - v^T a - h^T b, \tag{4}$$

where $\theta = \{W, L, R, c, b\}$ is the parameter of the BM model; W, L, R are the connection weights between nodes respectively, and the diagonal elements of L and R are 0. a and b represent the bias of v and h. Through this energy function, the probability distribution can be obtained by formula (1), and the solution of the model can be obtained by further solving. Although BM has a strong self-learning ability and can learn complex internal features in data, BM has a



(a)

(b)

Figure 1: Structural comparison of BM and RBM models. (a) Structure of the BM model. (b) Structure of the RBM model.

complex structure, resulting in a very long training time. In addition, it is difficult to obtain random samples of the distribution represented by BM, so the practical value is relatively low.

The difference between RBM and BM lies in that the neurons at the same layer are independent of each other, that is, $L = 0$ and $R = 0$. Only interlayer neurons are connected, and their structure is shown in Figure 1(b). Similarly, it includes a visible layer v and a hidden layer h. The visible layer mainly describes the features of the observed data, while the hidden layer serves as the feature extraction layer. If v includes n nodes $v = \{v_1, \dots v_n\}$, and h includes m nodes $h = \{h_1, \dots h_m\}$, then the energy function under a given set of states can be expressed as follows:

$$E(v, h; \theta) = -v^T W h - v^T a - h^T b = -\sum_{i=1}^{n}\sum_{j=1}^{m} v_i W_{ij} h_j - \sum_{i=1}^{n} a_i v_i - \sum_{j=1}^{m} b_j h_j, \tag{5}$$

where $\theta = \{W_{ij}, a_i, b_j\}$, W_{ij} is the weight matrix among the visible layer and the hidden layer. The purpose of learning RBM is to fit the distribution of training data by finding the appropriate parameter θ. In order to get the optimal value of θ, we can use the stochastic gradient ascent method. Therefore, the key step is to find the partial derivatives of each parameter. The gradient of RBM logarithmic likelihood function is as follows:

$$\frac{\partial L(\theta)}{\partial \theta} = \sum_{t=1}^{T}\left(\langle \frac{\partial(-E(v^{(t)}))}{\partial \theta} \rangle_{P(h|v^{(t)},\theta)} - \langle \frac{\partial(-E(v, h; \theta))}{\partial \theta} \rangle_{P(v,h;\theta)}\right). \tag{6}$$

In the above formula, $L(\theta)$ is the likelihood function of the RBM model, and $\langle \cdot \rangle_P$ represents the expectation of distribution P. For the former term, the probability distribution of h under a given sample can be calculated; for the latter term, all possible values of v need to be searched before the joint probability distribution can be calculated. Therefore, a feasible sampling method is needed to obtain the value of the distribution.

3.2. Gibbs Sampling Method. Gibbs Sample [1] is a sampling method based on the Markov Chain Monte Carlo (MCMC) strategy that constructs random samples of probability distributions of multiple variables. For example, the joint distribution of more than two variables is constructed in order to work out integrals and expectations. The efficiency of the MCMC algorithm is low because the high-dimensional data has a certain reception rate. If the reception rate can be set to 1, the problem of slow convergence caused by the frequent rejection of transfer can be avoided, and Gibbs sampling can sample the joint distribution of high-dimensional random variables. The specific process mainly, assuming a kd random vector $X = \{x_1, x_2, \dots, x_M\}$, cannot obtain the joint probability distribution $P(X)$ of X, but the rest of the components $x_{k^-} = \{x_1, x_2, \dots, x_{k-1}, x_{k+1}, \dots, x_M\}$ of a given X, the conditional probability of the k-th component x_k is $P(x_k|x_{k^-})$, therefore can from an initial state of$X$ (such as $[x_1^{(0)}, x_2^{(0)}, \dots, x_M^{(0)}]$), using the amount of conditional probability, iteratively to state the weight of samples, The distribution of the random variable converges geometrically to $P(X)$ as the number of samples increases.

Gibbs algorithm is employed to get random data conforming to the model distribution in the RBM model. The sampling process is introduced in Figure 2.

The specific steps of t-step Gibbs sampling in RBM are as three steps as follows:

Step 1: First, use the input sample to initialize the state v_0 of the visual node;

Step 2: Then, determine the sampling times t. Sampling is carried out according to the following conditional probability formula:

$h^{(s-1)}$ is obtained by sampling with conditional probability $P(h|v^{(s-1)})$;

Then the conditional probability $P(h|v^{(s-1)})$ is sampled to get v^s;

Among them, $s = 1, 2, \dots, t$.

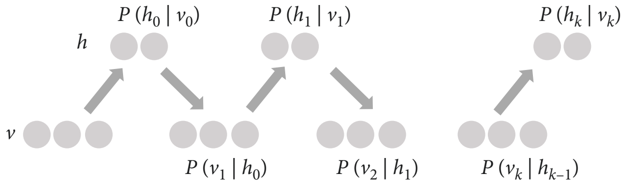


Figure 2: Gibbs sampling process.

Step 3: Step 2 is cyclically sampled for t times, and finally, when the sampling times t is large enough, v^t can be obtained.

3.3. Deep Belief Network Classification Method for Nonlinear Correction

3.3.1. DBN Model Based on Modified Elliott Function. DBN model is composed of multiple RBM stacked on top of each other, so in the training process of RBM, the activation function also determines the ability of feature extraction. RBM performs a step sampling by CD algorithm and Gibbs sampling. Firstly, the visible layer is mapped to the output of the hidden layer through the activation function, and then the output is taken as the input of the visible layer.

According to the analysis of the activation function in this article, it can be known that the activation function is a core position in network training. If the activation function is improperly selected, it is difficult to improve the accuracy of training learning no matter how to construct the model structure. However, if the activation function is properly selected, the feature extraction ability of the network can be significantly improved. Based on this, a DBN model based on the Modified Elliott function (ME-DBN) is proposed in our article. Elliott function [2] satisfies the generalized Logistic differential equation, so this paper introduces the Elliott function into the model to improve the traditional sigmoid activation function, as shown below:

$$f(x) = \frac{0.5x}{1+|x|} + 0.5. \tag{7}$$

In order to ensure that all neurons are saturated in the pretraining stage, the activation function should have a high gradient zero value. Based on this analysis, formula (7) is revised in this paper:

$$f(x) = \frac{0.5x}{\sqrt{1+x^2}} + 0.5. \tag{8}$$

Figure 3 shows the function graph of the modified Elliott function and sigmoid function.

As we can find from Figure 3, the modified Elliott function becomes steeper near zero, which causes more major features to fall into the middle region of the function, and at the same time, reaches the threshold at the lower value of its input, closer to the biological neuron than the sigmoid function.

Next, this paper compares the modified Elliott function with the sigmoid function, as shown in Figure 4:

As we can find from Figure 4 that ReLU has no gradient at the negative half-axis of input, and the modified function in this paper has a gradient, so the problem of failing to update the weight will not occur.

To better fit the distribution of input data in the network model in the pretraining stage, the activation function in the pretraining stage is improved in this paper. Therefore, after introducing the modified Elliott function into RBM, the conditional probability of the visible layer and hidden layer can be deduced as follows:

$$\begin{aligned} P(v_i = 1|h,\theta) &= \prod_i P(v_i|h)P(v_i = 1|h) = ME\left(\sum_j W_{ij}h_j + a_i\right) = \frac{\sum_j W_{ij}h_j + a_i}{2\sqrt{1+\left(\sum_j W_{ij}h_j + a_i\right)^2}} + \frac{1}{2},\\ P(h_i = 1|v,\theta) &= \prod_i P(h_i|v)P(h_i = 1|v) = ME\left(\sum_j W_{ij}v_j + b_i\right) = \frac{\sum_j W_{ij}v_j + b_i}{2\sqrt{1+\left(\sum_j W_{ij}v_j + b_i\right)^2}} + \frac{1}{2}. \end{aligned} \tag{9}$$

In the pretraining stage, the training objective is still the maximized likelihood function. CD algorithm is used for sampling, so the parameter update formula is as follows:

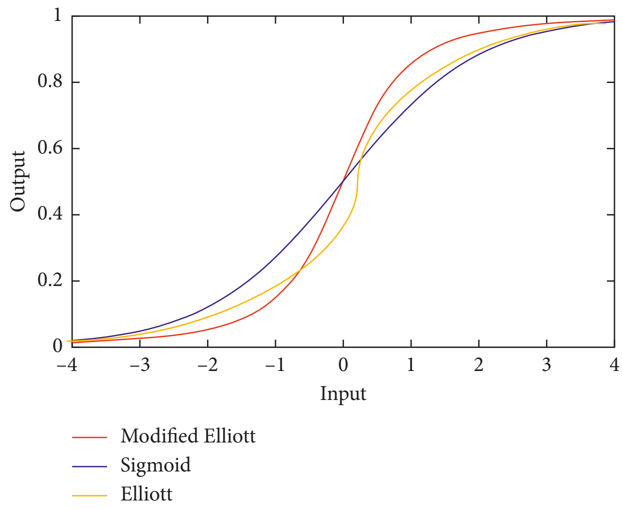


Figure 3: Comparison of modified Elliott function and sigmoid activation function.

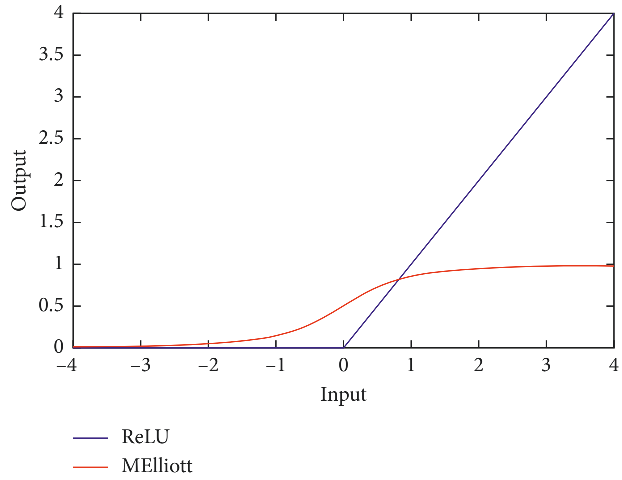


Figure 4: Comparison of modified Elliott function and ReLU activation function.

$$
\begin{aligned}
\frac{\partial L(\theta)}{\partial W_{ij}} &= \sum_{t=1}^{T}\left[P\left(h_j = 1|v^{(t)}\right)v_i^{(t)} - \sum_{v} P(v)P\left(h_j = 1|v\right)v_i\right] \\
&= \sum_{t=1}^{T}\left[\left(\frac{\sum_j W_{ij}v_i^{(t)} + b_j}{2\sqrt{1+\sum_j W_{ij}v_i^{(t)} + b_j^2}} + \frac{1}{2}\right)v_i^{(t)} - \sum_{v} P(v)\left(\frac{\sum_j W_{ij}v_i + b_j}{2\sqrt{1+\sum_j W_{ij}v_i + b_j^2}} + \frac{1}{2}\right)v_i\right], \\
\frac{\partial L(\theta)}{\partial a_i} &= \sum_{t=1}^{T}\left[v_i^{(t)} - \sum_{v} P(v)v_i\right], \\
\frac{\partial L(\theta)}{\partial b_j} &= \sum_{t=1}^{T}\left[P\left(h_j = 1|v^{(t)}\right) - \sum_{v} P(v)\left(h_j = 1|v\right)\right] \\
&= \sum_{t=1}^{T}\left[\left(\frac{\sum_j W_{ij}v_i^{(t)} + b_j}{2\sqrt{1+\sum_j W_{ij}v_i^{(t)} + b_j^2}} + \frac{1}{2}\right) - \sum_{v} P(v)\left(\frac{\sum_j W_{ij}v_i + b_j}{2\sqrt{1+\sum_j W_{ij}v_i + b_j^2}} + \frac{1}{2}\right)\right].
\end{aligned}
\tag{10}
$$

The parameters trained in the pretraining stage are used as the initialization of the fine-tuning stage, and the whole MEDBN network is fine-tuned by using the gradient descent method.

4. Evaluation Processing

After the training of these three classification models and training data, the performance of these models is evaluated with test data. The classifier best suited to the current traffic environment is that it has the most accurate classification model. Accuracy is represented as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}. \tag{11}$$

In the formula, TP is a true positive, indicating that the traffic that belonged to category C is classified in category C. FP is a false positive, showing that the traffic not belonging to category C is classified by mistake; FN is missing report, indicating that the traffic not belonging to category C is classified into others; TN is a true negative, indicating that the traffic of noncategory C that is classified as noncategory C.

The precision defined in formula (11) is used to select the optimal proposed model. At the same time, three indicators are used to evaluate the performance of the proposed model, which are Precision, Recall, and F1 score, The definition is as follows:

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP + FP}, \\ \text{Recall} &= \frac{TP}{TP + FN}, \\ F1_{\text{score}} &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \end{aligned} \tag{12}$$

5. Data Processing

ISCX VPn-NonVPN Traffic Dataset was selected in the experiment. As shown in Table 1, this data set consists of 15 applications, such as Facebook, Youtube, Netflix, and so on. The selected application uses various security protocols for encryption, including HTTPS, SSL, SSH, and proprietary protocols. The selected data set contains a total of 206,688 packets. Obviously, the data set is unbalanced. Some applications have a large number of traffic samples, such as Netflix, which accounts for 25.126% of the total data set. Meanwhile, some applications have a small number of traffic samples, such as Aim Chat and ICQ, which only account for 2-3% of the total data set.

Table 1: Description of sample dataset.

The application name	Unbalanced sample	
	Quantity	Ratio (%)
AIM_chat	4869	2.3
SCPdown	15390	7.4
Youtuebe	12738	6.1
Voipbuster	35469	17.1
E-mail	4417	2.1
Vimeo	18755	9
Facebook	5527	2.6
Gmail	7329	3.5
TorTwitter	14654	7
Hangout	7587	3.6
Spotify	14442	6.9
Skype	4607	2.2
ICQ	4243	2
SETPdown	4729	2.2
Netflix	51932	25.1
Total	206688	100

6. Experiment and Simulation

To further distinguish the effectiveness of our proposed model for identifying encrypted traffic, we list a series of classic benchmark models that have been proven to achieve excellent prediction and classification results in various fields. Figure 5 shows the evaluation of classification recognition results with the benchmark models. They include XGboost algorithm and GBDT algorithm based on number model, Bayesian classification algorithm and SVM algorithm based on classical classification algorithm, LSTM model and RNN model based on the neural grid. At the same time, we also take a single DBN model as one of the benchmark models to compare the classification recognition results between the DBN model and our ME-DBN model. As can be seen from the figure above, (1) Among all benchmark models, the ME-DBN model achieves the best performance in five indicators, which indicates that our proposed model is effective in the classification of encrypted traffic; (2) compared with all benchmark models, DBN model achieves the best results in ACC and F1 indicators, and also ranks TOP3 in classification results of other indicators, which indicates that, on the whole, DBN model can effectively identify encrypted traffic; (3) although RNN model achieves the best result in FRrate index, its performance in other indicators is poor. We could find that classification results based on the RNN model are unstable. (4) Compared with the DBN model, the performance of ME-DBN proposed by us is superior to the DBN model in all five indicators, which indicates that the method proposed by us can effectively enhance the basic DBN model; (5) we can also know from the experimental results that neural grid based models such as CNN, RNN, and LSTM have significantly higher classification and identification effects on encrypted traffic than other benchmark models (including decision tree-based classification model and classical mathematical classification model).

Figures 6 and 7 show the comparison of training and prediction time of different algorithms on the ISCX VPN-nonVPN dataset. Among them, 70% samples are selected as training samples and 30% samples are selected as test samples. It can be seen from the figure that in the ISCX

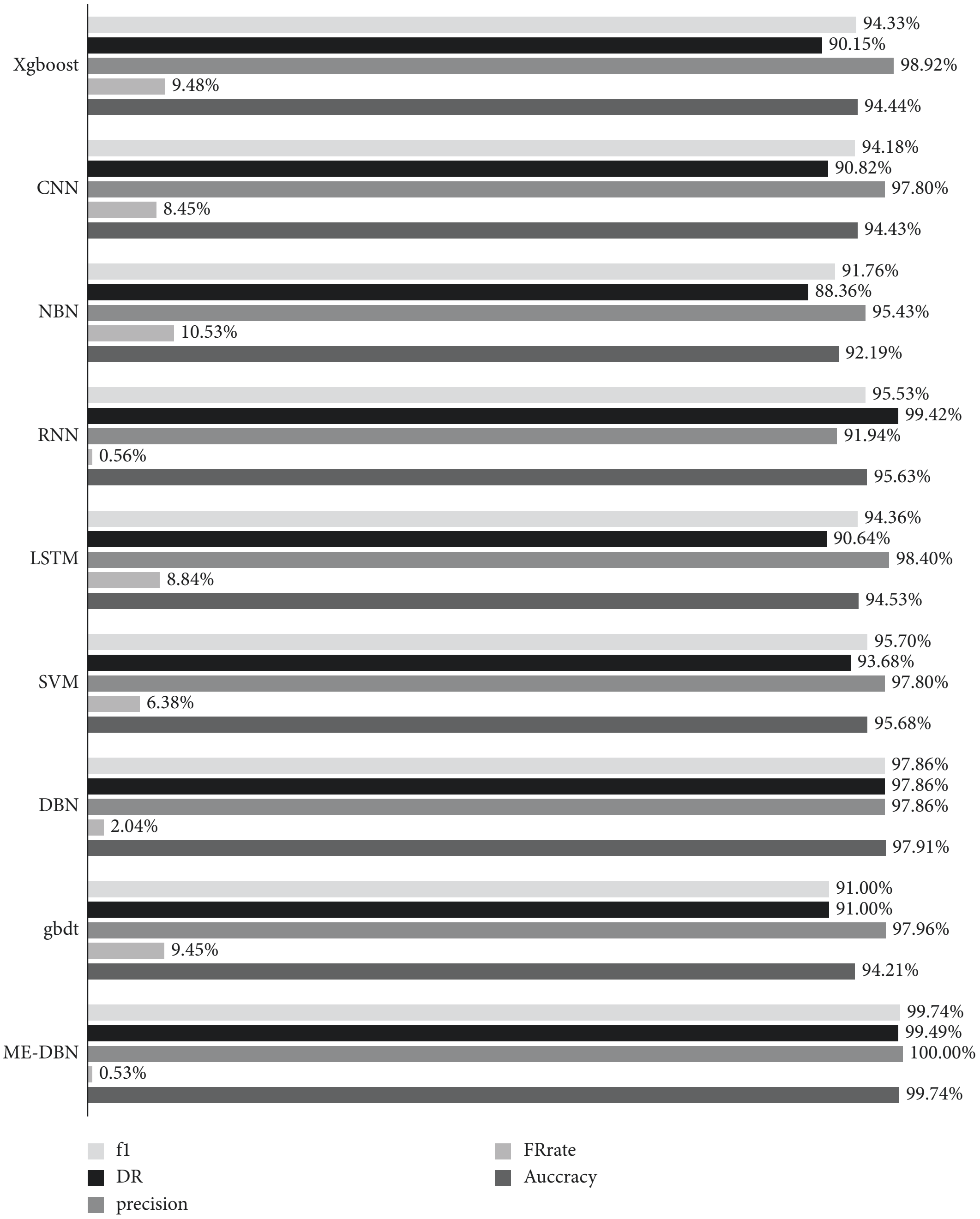


Figure 5: Evaluation of classification recognition results with the benchmark models.

VPN-NonVPN data set, the CNN model has the longest training time and the NBN model has the shortest prediction time, but at the same time, the CLASSIFICATION accuracy of the NBN model is not high. ME-DBN has more training time on the dataset than the DBN model because sparse regular terms are added to the likelihood function and the derivation process affects the training time. For deeper structure, it needs more training time than the traditional SVM and XGboost model because of the complicated calculation process, but the classification results have a certain increase, and the algorithm proposed in our article the ME - DBN and DBN algorithm forecast time differs only 4 s, demonstrate the algorithm based on sparse model time performance improves the classification accuracy. It has better classification performance.

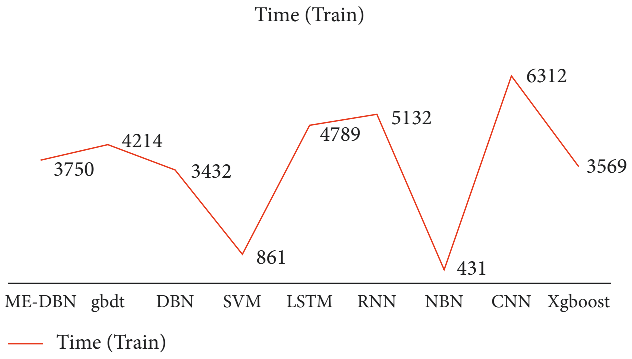


Figure 6: Comparison of training time of different algorithms.

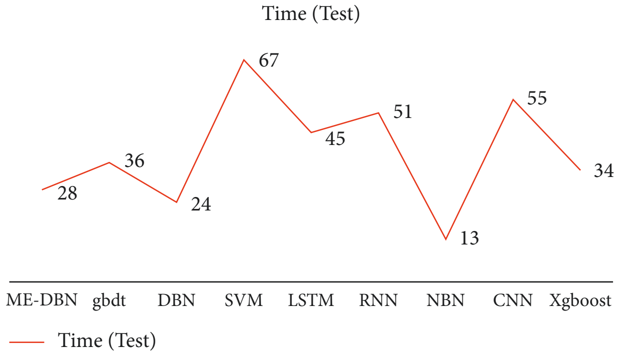


Figure 7: Comparison of prediction time of different algorithms.

7. Conclusions

At present, the deep neural network has become an important research content on machine learning. Feature extraction algorithm based on DNN mainly uses a deep neural network model to carry out feature extraction of data imitating the information processing mechanism of the human brain, so as to screen the important information in data. The deep neural network has an excellent performance in extracting images, sound, text, and other information. However, with the increasing scale of data sets, the network structure becomes more and more complex, making network training more difficult, which requires more effective training methods. Secondly, when the traditional sparse deep network model learns input data, all hidden layer nodes may have the same effect without completely changing the feature homogeneity phenomenon. In addition, the traditional Sigmoid activation function is nonzero mean, which is difficult to effectively train the network and is prone to the phenomenon of gradient disappearance. To handle the above problems, our article studies the feature extraction algorithm based on DNN, and proposes a DBN traffic classification method based on the nonlinear correction. In view of the phenomenon of gradient disappearance that the traditional Sigmoid function is prone to, the Elliott function satisfying the generalized Logistic differential equation is proposed to replace the Sigmoid activation function, and then the Elliott function is modified to meet the characteristics of RBM. The modified Elliott function can make the nodes in the saturation state, so it is not easy to cause the problem of gradient disappearance.

Data Availability

The support data can can be obtained from the author upon request.

Conflicts of Interest

The author declares no conflicts of interest.

References

[1] S. Kumar, S. Dharmapurikar, F. Yu, P. Crowley, and J. Turner, “Algorithms to accelerate multiple regular expressions matching for deep packet inspection,” *ACM SIGCOMM - Computer Communication Review*, vol. 36, 2006.

[2] Y. Zeng, M. Qiu, Z. Ming, and M. Liu, “Senior2Local: a machine learning based intrusion detection method for VANETs,” in *Proceedings of the International Conference on Smart Computing and Communication*, December 2018.

[3] M. Shen, M. Wei, L. Zhu, and M. Wang, “Classification of encrypted traffic with second-order Markov chains and application attribute bigrams,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1830–1843, 2017.

[4] C. V. Wright, F. Monrose, and G. M. Masson, “On inferring application protocol behaviors in encrypted network traffic,” *Journal of Machine Learning Research*, vol. 7, pp. 2745–2769, 2006.

[5] W. Wang, Y. Sheng, J. Wang et al., “HAST-IDS: learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection,” *IEEE Access*, vol. 6, pp. 1792–1806, 2018.

[6] M. Lotfollahi, R. S. H. Zade, M. J. Siavoshani, and M. Saberian, “Deep packet: a novel approach for encrypted traffic classification using deep learning,” 2017, https://arxiv.org/abs/1709.02656.

[7] J. Hochst, L. Baumgartner, M. Hollick, and B. Freisleben, “Unsupervised traffic flow classification using a neural autoencoder,” in *Proceedings of the IEEE 42nd Conf. Local Comput. Netw. (LCN)*, pp. 523–526, Singapore, October 2017.

[8] P. Wang, F. Ye, X. Chen, and Y. Qian, “Datanet: deep learning based encrypted network traffic classification in sdn home gateway,” *IEEE Access*, vol. 6, pp. 55380–55391, 2018.

[9] M. A. Qatf, Y. Lasheng, M. A. Habib, and K. A. Sabahi, “Deep learning approach combining sparse autoencoder with SVM for network intrusion detection,” *IEEE Access*, vol. 6, pp. 52843–52856, 2018.

[10] K. Claffy, H. Braun, and G. Polyzos, “A parameterizable methodology for internet traffic flow profiling,” *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, pp. 1481–1494, 1995.

[11] R. Gu, H. Wang, Y. Sun, and Y. Ji, “Fast traffic classification using joint distribution of packet size and estimated protocol processing time,” *IEICE - Transactions on Info and Systems*, vol. 93, no. 11, pp. 2944–2952, 2010.

[12] S. Yeganeh, M. Eftekhar, Y. Ganjali, R. Keralapura, and A. Nucci, “CUTE: traffic classification using terms,” in *Proceedings of the 2012 21st International Conference on Computer Communications and Networks ICCCN*, pp. 1–9, Munich, Germany, July 2012.

[13] H. Ibrahim, S. Nor, and H. Jamil, “Online hybrid internet traffic classification algorithm based on signature statistical and port methods to identify internet applications,” in *Proceedings of the 2013 IEEE International Conference on Control*

System, Computing and Engineering, pp. 185–190, Penang, Malaysia, November 2013.

[14] M. Conti, L. Mancini, R. Spolaor, and N. Verde, “Can’t you hear me knocking: identification of user actions on android apps via traffic analysis,” in *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, pp. 297–304, New York, NY, U.S.A, March 2015.

[15] G. Xiong, W. Huang, Y. Zhao, M. Song, Z. Li, and L. Guo, *Real-time Detection of Encrypted Thunder Traffic Based on Trustworthy Behavior Association*, Springer, Berlin Germany, pp. 132–139, 2013.

[16] A. McGregor, M. Hall, P. Lorier, and J. Brunskill, *Flow clustering using machine learning techniques*, Springer, Berlin, Germany, pp. 205–214, 2004.

[17] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, “Malware traffic classification using convolutional neural network for representation learning,” in *Proceedings of the 2017 International Conference on Information Networking (ICOIN)*, pp. 712–717, IEEE, Da Nang, Vietnam, January 2017.

[18] M. Lotfollahi, M. J. Siavoshani, R. S. H. Zade, and M. Saberian, “Deep packet:A novel approach for encrypted traffic classification using deep learning,” *Soft Computing*, vol. 24, no. 3, pp. 999–2012, 2020.

[19] L. Grimaudo, M. Mellia, and E. Baralis, “Hierarchical learning for fine grained intemet traffic classification,” in *Proceedings of the Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 463–468, IEEE, Limassol, Cyprus, August 2012.

[20] M. Lotfoilahi, R. S. H. Zade, M. J. Siavoshani, and M. Saberian, “Deep packet:A novel approach for encrypted traffic classification using deep learning,” *Soft Computing*, vol. 24, no. 3, pp. 1999–2012, 2017.

[21] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[22] T. Qin, L. Wang, Z. Liu, and X. Guan, “Robust application identification methods for P2P and VoIP traffic classification in backbone networks,” *Knowledge-Based Systems*, vol. 82, pp. 152–162, 2015.

[23] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescape, “Mobile encrypted traffic classification using deep learning,” in *Proceedings of the 2018 Network Traffic Measurement and Analysis Conference (TMA)*, pp. 1–8, Vienna, Austria, June 2018.

[24] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescape, “Mobile encrypted traffic classification using deep learning: experimental evaluation, lessons learned, and challenges,” *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 445–458, 2019.

[25] M. Martín, B. Carro, A. S. Esguevillas, and J. Lloret, “Network traffic classifier with convolutional and recurrent neural networks for internet of things,” *IEEE Access*, vol. 5, pp. 18042–18050, 2017.

[26] J. Hochst, L. Baumgartner, M. Hollick, and B. Freisleben, “Unsupervised traffic flow classification using a neural autoencoder,” in *Proceedings of the 2017 IEEE 42nd Conference on Local Computer Networks (LCN)*, pp. 523–526, Singapore, October 2017.