Applied Mathematics, Modeling and Computer Simulation
C.-H. Chen et al. (Eds.)

doi:10.3233/ATDE221055

A Deep Learning-Based Model for Classifying Malicious Network Traffic

Liang GU[1], Ting DI, Xin ZHOU
State Grid Shanxi Province Electric Power Company Information and Communication Branch, Taiyuan 030000, China

Abstract. Network traffic, as a carrier of information transmission and interaction, portrays the behavior trajectory of users and functions as an important approach to detect network attacks and analyze network anomalies. To address the problems such as low recognition rate, high false alarm rate and inability to detect unknown traffic due to the difficulty of traffic feature extraction, over-reliance on manual experience and feature techniques in traditional network traffic analysis practice, this paper proposes a network traffic classification method that integrates mutual information and convolutional neural network, which, not relying too much on manual feature extraction, assigns weights to network features by mutual information and accomplishes the classification and analysis of malicious traffic based on convolutional neural network and short and long-term memory network. The experimental results show that the method can greatly improve the detection rate of malicious traffic with excellent robustness and generalization while reducing the artificial dependence.

Keywords. Traffic Classification; Mutual Information Theory; Convolutional Neural Network

1. Introduction

The Internet has played a key role in building new social infrastructure, interrupting epidemics, and transforming social development patterns and human lifestyles. Due to the trend of people's increasing reliance on the Internet, network-based attacks are also emerging, and strengthening network security research and enhancing network attack detection capabilities remain a hot issue in current network security research.

Network traffic analysis defines traffic other than normal business as abnormal traffic, and attack traffic is a kind of abnormal traffic, traffic analysis is a method to detect network attacks from the traffic. Shen et al [1] used graphical neural networks to classify traffic by collecting application traffic and analysing its characteristics to build a traffic interaction graph. However, the applicability of this method in the current mass application scenario is challenging because it requires the collection of application traffic for annotation. With the prevalence of traffic encryption and privacy protection technologies, traditional traffic analysis techniques have become difficult to cope with. Zheng et al [2] proposed an end-to-end traffic-based relational network (RBRN) for classification of encrypted traffic using meta-learning techniques, and

[1] Corresponding Author, Liang GU, State Grid Shanxi Province Electric Power Company Information and Communication Branch, Taiyuan 030000, China; E-mail: 740304606@qq.com.

experiments showed that its generalisation was good. The resource consumption of post-decryption detection methods in encrypted traffic analysis is high; non-decryption detection relies on machine learning techniques with strong reliance on manual features; deep learning methods can effectively solve the problems faced by post-decryption detection by learning the non-linear relationship between the original input and the corresponding output. To address the problem that the above-mentioned research methods only classify but not identify attack traffic and the problem that feature labels are outdated and plague traffic detection accuracy, Zeng et al [3] proposed a deep learning-based intrusion detection framework for encrypted traffic called DFR (deep-full-range), which has an average score of 12.15% in intrusion detection F1. Yakubu et al [4] proposed a bi-directional long and short-term memory network-based classifier (Bi-LSTM) for the problem of low classification accuracy, and its classification accuracy reached 92.81%; Shi Leyi et al [5] proposed an intrusion classification model based on correlation information entropy and CNN-BILSTM for the problem of difficult feature extraction in industrial control systems, and the accuracy of this model reached 99.21%.

In view of the increasing threat of network intrusion and the inability of features to be incrementally updated for attack variants, this paper proposes for the first time a classifier for malicious network traffic based on mutual information theory, convolutional neural networks and deep learning techniques such as long and short term memory networks, which can optimally extract effective feature selection, fully learn network spatial and temporal features, significantly improve the detection rate of network intrusion, and has good generalization.

The paper is organised as follows: Section 1 introduces the basic theory of classifier construction, Section 2 discusses the classifier construction method, Section 3 validates the performance of the classifier, and Section 4 summarises the research work.

2. Basic Theory

2.1. Mutual Information Theory

Mutual Mutual Information (MI)[6] is used to measure the degree of interdependence between two variables, the larger the MI value, the higher the correlation between the two. The principle is shown in equation (1), where H(X), H(Y) denote the information entropy of variable X, variable Y respectively, and its principle is shown in equation (2); H(X,Y) denotes the cross-entropy of variable X and variable Y, and its principle is shown in equation 3.

$$\begin{aligned} I(X;Y) &= H(X) - H(X \mid Y) = H(X) + H(Y) - H(X,Y) \\ &= \sum_{x \in X} P(x) \log \frac{1}{p(x)} + \sum_{y \in Y} P(y) \log \frac{1}{p(y)} + \sum_{x \in X} \sum_{y \in Y} P(x,y) \log \frac{1}{p(x,y)} \\ &= \sum_{x \in X} \sum_{y \in Y} P(x,y) \log \frac{p(x,y)}{p(x)p(y)} \end{aligned} \tag{1}$$

$$H(X) = \sum_{x \in X} p(x) \log p(x) \tag{2}$$

$$H(X,Y)=\sum_{x\in X}\sum_{y\in Y}p(x,y)\log\frac{1}{p(x,y)} \tag{3}$$

2.2. CNN Networks

CNN was proposed by Hubel and Wiesel [7], and Fukushima K [8]proposed CNN model based on it, which is a network model integrating automatic feature extraction, feature selection and classification. CNN network effectively extracts local features of data through convolution and pooling operations, and in the fully connected layer through The CNN network is designed to achieve malicious/normal traffic binary classification by efficiently extracting local features through convolution and pooling operations, and mapping them to probability values in the range of [0,1] using softmax activation functions in the fully connected layer.

2.3. Bidirectional Long Short Term Memory

Bidirectional Long Short Term Memory (Bi-LSTM) is a modification of LSTM [9], which is improved by training 2 copies on the original dataset, an LSTM trained on the initial dataset and an LSTM trained on the reverse copy dataset of the original input data, which has a better learning ability for features with temporal The LSTM has a better learning ability for features with temporal sequential characteristics.

The data flow diagram of the LSTM model is shown in figure 1, which mainly consists of cell and gate control. cell is similar to the memory of the network during the whole data processing process, which is used to save and transmit relevant information, and gate control mainly includes forgetting gate, input gate, current state control gate and output gate. The forgetting gate mainly selectively forgets the input passed in from the previous node, i.e. forgetting the unimportant ones and remembering the important information; the input gate and the current state control gate decide how much of the input of the network is saved into the cell at the current moment; the output gate decides how much of the cell of the cell state is output into the current output value of the LSTM.

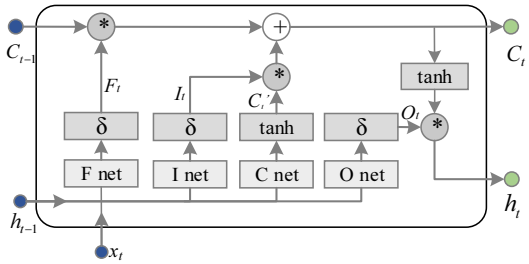

Figure 1. The operation schematic of Convolution

F net, I net, C net and O net in figure 1 denote the forgetting gate, input gate, current state control gate and output gate respectively. The principle of forgetting gate and input gate is shown in equation (4)(5). w_f in equation (4) is the weight matrix of the forgetting gate, $[h_{t-1},x_t]$ denotes the vector that connects the two vectors for a longer time, bf is the bias term of the forgetting gate and is a sigmoid function.

$$F_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f) \tag{4}$$

$$I_t = \delta(w_i \cdot [h_{t-1}, x_t] + b_i) \tag{5}$$

The new cell state *Ct* at the current moment is related to both information from a long time ago and information from the previous moment, so its calculation requires consideration of combining the current input memory *Ct'* with the long-term memory *Ct-1* to form the new cell state *Ct*. The new cell state *Ct* is calculated as shown in equation (6), where *Ct '* is calculated as shown in equation (7).

$$C_t = F_t \cdot C_{t-1} + I_t \cdot C_t^{'} \tag{6}$$

$$C_t^{'} = \tanh(w_c \cdot [h_{t-1}, x_t] + b_c) \tag{7}$$

ht indicates that the output gate uses the tanh function to control how much of the new cell state *Ct* is output to the next node, and *Ot* is the output to the current node state. *Ot* is calculated as shown in equation (8) and the output gate is calculated as shown in equation (19).

$$O_t = \delta(w_o \cdot [h_{t-1}, x_t] + b_o) \tag{8}$$

$$h_t = O_t \cdot \tanh(C_t)) \tag{9}$$

The Bi-LSTM model is the result of stitching two LSTM sequences together for output, and its model structure is shown in figure 2. The result can be expressed in Eq. (10),where $\overrightarrow{h_t}$, $\overleftarrow{h_t}$ denote the output of the rightward LSTM and leftward LSTM, respectively, and b is the bias term.

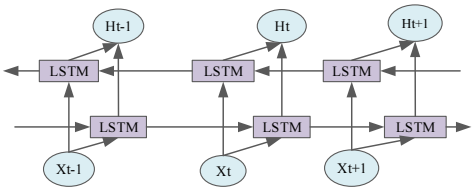

Figure 2. The structure of of Bi-LSTM model

$$y_t = w_r \overrightarrow{h_t} + w_l \overleftarrow{h_t} + b \tag{10}$$

3. MCBL-based Traffic Classifiers

3.1. Data Preprocessing

Since the data set used for the simulation [10] has the problem of non-unique data types and inconsistent dimensions, the data is pre-processed by binary conversion, normalization, and One-Hot coding in turn. The dichotomous conversion was carried out by using dictionary key-value pairs to convert labels to values, converting "Benign" to 0 and all other attack types to 1. The data were then normalised to fall between [0, 1] to reduce training time, as shown in equation (11), where x represents the actual value of the sample， $x_{\min}, x_{\max}$ denote the minimum and maximum values of the sample values respectively. Finally the classification values are processed using One-Hot coding [11], which in turn ensures that the distance between features is calculated more reasonably.

$$X = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{11}$$

3.2. MCBL Classifier Build Process

The MCBL-based traffic classifier consists of two main stages: (1) feature selection for raw network traffic using the MI feature selection module; and (2) construction of a CBL classifier for training to achieve malicious/normal traffic classification. the flowchart of the MCBL traffic classifier is shown in figure 3.

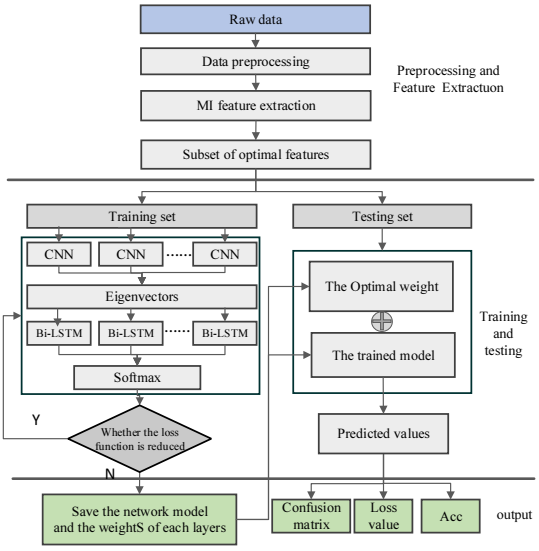

Figure 3. The flow chart of MCBL traffic classification model

The main steps of the MCBL flow classifier include:

Step1: Data pre-processing.

Step2: Feature selection based on MI theory to eliminate redundant and invalid variables from the data.

Step3: splitting the data samples into training and test sets.
Step4: Training the CBL-based classifier.
Step5: Use the test set for model evaluation.

3.3. Feature Selection Based on MI Theory

Each feature in the dataset has a different degree of correlation with the labels. In this paper, we use MI theory to represent the contribution of each feature to the labels; the higher the MI value, the higher the contribution, and the better for model classification. there are 16 features with MI values over 0.2 and 49 features over 0.1, of which 10 features with MI values of 0 are invalid features, and their existence not only does not improve the model accuracy, but also causes a lot of time wastage. In this paper, in order to fully learn the original data features and effectively improve the model detection performance, the features with MI values over 0.1 are selected as the final input variables of the model..The features with mutual information value greater than 0.1 selected in this paper are shown in table 1.

Table 1. The Mutual information value between each feature and label

Column number	MI	Column number	MI	Column number	MI
70	0.785092	55	0.193750	3	0.16267
0	0.483325	7	0.191635	18	0.16207
68	0.483060	66	0.191635	17	0.161208
67	0.457895	56	0.187231	1	0.150957
37	0.414168	14	0.187231	53	0.140263
36	0.385082	50	0.186429	31	0.12647
4	0.263750	39	0.184819	49	0.106998
63	0.263750	15	0.182929	26	0.105893
5	0.261670	43	0.178371	22	0.105874
65	0.261670	44	0.178196	23	0.105558
69	0.259565	21	0.177426	2	0.102588
41	0.230970	48	0.·117530	16	0.102588
8	0.230295	54	0.172590	25	0.102131
12	0.223594	42	0.172242	30	0.100888
6	0.204824	11	0.166873	27	0.100164
64	0.204824	20	0.163464		
10	0.193750	38	0.163237		

3.4. Performance Assessment

The MCBL classifier performance evaluation metrics include confusion matrix, accuracy, detection rate and training time. The confusion matrix is used to represent the matrix of a classifier result, and its matrix representation is shown in table 2, where TP denotes true positive rate, FN denotes false negative rate, FP denotes false positive rate and TN denotes true negative rate. The accuracy and detection rates are calculated in equations (12) and (13).

Table 2. The Confusion Matrix

Real Tags / Predicted results	Normal	Abnormal
Normal	TP	FN
Abnormal	FP	TN

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

$$recall = sensitivit\,y = DR = \frac{TP}{TP + FN} \tag{13}$$

4. Simulation Experiments and Analysis

In this paper, the CSE-CIC-IDS 2018 dataset [10] was selected for experimentation, which includes three types of data, Benign, SSH-Bruteforce, and FTP-BruteForce, and the data distribution is shown in table 3. In this paper, the Benign traffic data in the dataset is recorded as "normal" and the other traffic data is recorded as "abnormal".

Table 3. The data type distribution details of CSE-CIC-IDS2018 dataset

The label type	Data volume	Percentage
Benign	667626	63.7%
SSH-Bruteforce	187589	17.9%
FTP-BruteForce	193360	18.4%

4.1. Impact of Feature Selection on Classifier Training

As the dataset [10] contains a large number of redundant variables and invalid variables, this paper uses MI for correlation analysis of each feature and label, selects strongly correlated features as the input of the model, and verifies the correctness of feature selection by comparing the experimental results before and after MI feature selection, the results are shown in table 4, the number in the model name is the number of features selected by mutual information, for example $49_{(0.1)}$ means 49-dimensional features with MI values greater than 0.1 were selected as input.

Table 4. The training results of features selected by different mutual information

Model name	Acc %	Recall %	Time/s
$MCBL16_{(0.2)}$	99.9981	99.9985	2431s
$MCBL49_{(0.1)}$	99.9986	99.9992	4519s
$MCBL79_{(0)}$	99.9979	99.9987	6961s

Table 4 shows that the model without feature selection has no advantage in training time and model detection performance; selecting features with MI values greater than 0.1 as input to the model can achieve higher detection performance; selecting features with mutual information values greater than 0.2 results in a significant reduction in model training time, but also a small reduction in detection performance. It can be found that MI-based feature selection can significantly compress the classifier training time with almost no impact on the classification progress.

4.2. MCBL Based Traffic Classifier

Since the Bi-LSTM module in MCBL mainly focuses on learning the time series characteristics of the original data, and the feature "Timestamp" in the original dataset

is numbered 2, i.e. the feature selection should include this column as much as possible, obviously the mutual information cannot be selected when it is greater than 0.2. At the same time, considering the training performance and training efficiency of the classifier, combined with the training time and classification performance results in table 4, priority was given to the classification performance, so the results with MI values greater than 0.1 were used as the final choice for comparison with other models in this paper, and the confusion matrix for the MCBL$(49)_{0.1}$ simulation experiment is shown in figure 4.

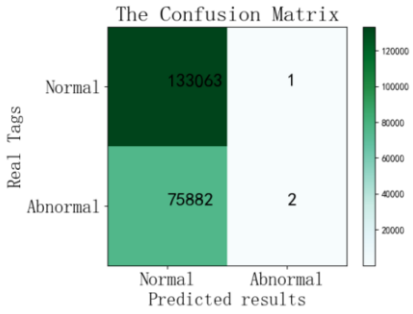

Figure 4. The confusion matrix of MCBL model

4.3. Comparative Experiments

To further validate the performance of the MCBL classifier, it was compared with the Bi-LSTM and CNN+Bi-LSTM models proposed in the literature [4] and [5], and the comparison results are shown in table 5 and figure 5, respectively.

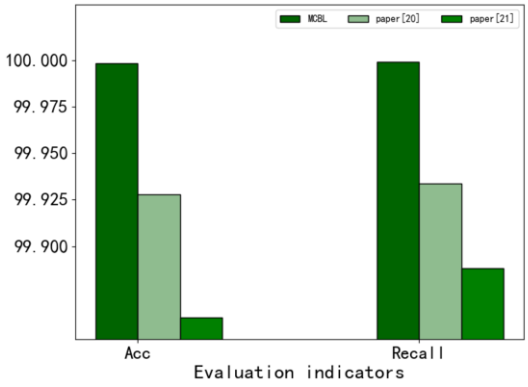

Figure 5. The Comparison of the performance of MCBL with the literature [4] and [5]

Table 5. The Comparative study of MCBL model and other models

Model name	Acc %	Recall %	Time/s
MCBL	99.9986	99.9992	4519s
CNN+Bi_LSTM[5]	99.9979	99.9987	6961s
Bi_LSTM [4]	99.8617	99.8880	8248s

As can be seen from table 5, the accuracy and detection rates of MCBL are higher than those of the models in [4] and [5], indicating that the model has better detection capability for malicious traffic, mainly because the CNN and Bi-LSTM networks based

on MI feature selection can learn the features of the original data more fully and improve the detection performance of the model; in addition, compared with the two models in [4] and [5], the MCBL classifier has significantly more advantages in terms of training time, mainly because MI can significantly reduce the dimensionality of the original data and shrink the training time.

5. Conclusions

Traffic classification is one of the main ways to detect network intrusion. In this paper, we propose a traffic classifier (MCBL) based on deep learning techniques, fusing CNN and Bi-LSTM. The simulation experiments based on the selected dataset show that MCBL has good classification performance, which is a significant improvement over existing studies.

Acknowledgment

Research supported by State Grid Shanxi electric power company science and technology project Foundation (No. 52051C21000G).

References

[1] Shen M, Zhang J, Zhu L, et al. Accurate decentralized application identification via encrypted traffic analysis using graph neural networks[J]. IEEE Transactions on Information Forensics and Security, 2021, 16: 2367-2380.
[2] Zheng W, Gou C, Yan L, et al. Learning to classify: A flow-based relation network for encrypted traffic classification[C]//Proceedings of The Web Conference 2020. 2020: 13-22.
[3] Zeng, Yi, et al. "Deep-Full-Range: a deep learning based network encrypted traffic classification and intrusion detection framework." IEEE Access 7 (2019): 45182-45190.
[4] Yakubu Imrana, Yanping Xiang, Liaqat Ali, Zaharawu Abdul-Rauf. A bidirectional LSTM deep learning approach for intrusion detection[J]. Expert Systems with Applications, 2021,185,115524.
[5] Shi Leyi,Zhu Hongqiang,Liu Yihao,Liu Jia. Intrusion detection of industrial control systems based on correlation information entropy and CNN-BiLSTM [J]. Computer Research and Development, 2019, 56(11):2330-2338.
[6] Debapriya Sengupta, Phalguni Gupta, Arindam Biswas, A survey on mutual information based medical image registration algorithms[J], Neurocomputing, 2022,486,174-188.
[7] HUBEL DH, WIESEL TN. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. J Physiol. 1962:160(1): 106-54.
[8] Fukushima K. Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol Cybern. 1980;36(4):193-202.
[9] Chunhe Song, Yingying Sun, Guangjie Han, Joel J.P.C. Rodrigues. Intrusion detection based on hybrid classifiers for smart grid[J], Computers & Electrical Engineering, 2021, 93, 107212.
[10] Canadian Institute for Cybersecurity, CSE-CIC-IDS 2018 dataset [EB/OL], University Of New Brunswick. [2021-12-22].https://www.unb.ca/cic/datasets/ids-2018.html.
[11] J. Liang, J. Chen, J. H. Zhang, X. Q. Zhou, Y. Zhou, J. J. Lin. Anomaly detection based on unique thermal coding and convolutional neural network[J]. Journal of Tsinghua University (Natural Science Edition), 2019,59(07):523-529.