# IoT Intrusion Detection System Based on LSTM Model

Weiqun Li[1], Chaowen Chang[1, *]

[1]*University of Information Engineering, Zhengzhou, China 450000*
*f_author: lwqun9080@163.com, s_author: changchaowen@hnsl.gov.cn*

**Abstract**

Aiming at the problems of time-consuming feature extraction and general efficiency in the detection of en-crypted traffic by traditional machine learning algorithms, an intrusion detection model based on deep learning long short-term memory network (LSTM) was proposed. First, the malicious encrypted traffic in the CTU-13 data set and the normal traffic in the CICIDS-2017 data set are extracted to form a data set; then the binary classification data set processing is completed based on the secure transport layer protocol; finally, the LSTM and one-dimensional convolutional neural networks are trained. Network, two-dimensional convolutional neural network and convolutional neural network-long short-term memory network four deep learning models. The experimental results show that LSTM has significant advantages over the other three models in five evaluation parameters, the accuracy of key parameters is as high as 99.84%, and it performs well in terms of CPU and memory usage, which meets the security requirements of the Internet of Things.

***Keywords:*** *LSTM; CTU-13; CICIDS-2017; TLS; CNN; Accuracy*

## 1 INTRODUCTION

Transport Layer Security (TLS) protocol brings security protection to IoT data transmission, while some malware hides malicious attack behavior through TLS protocol, which leads to traditional malware detection technology Invalidation based on Deep Packet Inspection (DPI) and signature or model matching. Intrusion Detection System (IDS) plays an important role in securing IoT networks by detecting and preventing malicious activities.

The goal of encrypted traffic anomaly detection is to discover hidden distributed denial of service (DDoS) attacks, botnets, Heartbleed vulnerabilities, and other malicious codes such as Trojan horses and worms hidden in the traffic. There are three difficulties in detection. Items: 1) Insufficient labeled samples. The training of classic machine learning models needs to rely on huge labeled training sets. Currently, public datasets for encrypted traffic detection are relatively scarce, and compared with normal traffic, the number of malicious traffic is far insufficient, making it difficult to train a better classification model. 2) High detection accuracy requirements. Massive IoT devices collect a lot of sensitive information, prompting models used to detect malicious encrypted traffic to improve detection accuracy and reduce the cost of false positives and false negatives. 3) Lack of feature information. The features of traditional anomaly detection, such as payload, packet length sequence, packet arrival time, etc., are difficult to identify malicious encrypted traffic, and the available feature information is relatively small.

The goal of encrypted traffic anomaly detection is to discover hidden distributed denial of service (DDoS) attacks, botnets, Heartbleed vulnerabilities, and other malicious codes such as Trojans and worms hidden in the traffic. There are three difficulties in detection. Items: 1) Insufficient labeled samples. The training of classic machine learning models needs to rely on huge labeled training sets. Currently, public datasets for encrypted traffic detection are relatively scarce, and compared with normal traffic, the number of malicious traffic is far insufficient, making it difficult to train a better classification model. 2) High detection accuracy requirements. Massive IoT devices collect a lot of sensitive information, prompting models used to detect malicious encrypted traffic to improve detection accuracy and reduce the cost of false positives and false negatives. 3) Lack of feature information. The features of traditional anomaly detection, such as payload, packet length sequence, packet arrival time, etc., are difficult to identify malicious encrypted traffic, and the available feature information is relatively small.

This paper constructs a data set of encrypted traffic, uses feature engineering to extract the feature information of the original network traffic of the data set,


B. Fox et al. (Eds.): IC-ICAIE 2022, AHCS 9, pp. 1404-1409, 2023.
https://doi.org/10.2991/978-94-6463-040-4_209

and uses the processed features as the training data set of the deep learning model. Use the deep learning algorithm of Long Short-Term Memory (LSTM) to complete the training classification, compare one-dimensional convolutional neural network (1D_CNN), two-dimensional convolutional neural network (2D_CNN) and convolutional neural network (2D_CNN) Neural Network-Long Short Term Memory Network (CNN_LSTM) algorithm, LSTM has better performance under this training dataset. The main contributions are as follows:

(1) Construct an encrypted data set, use data set CTU-13 to extract 20 different types of malware traffic, and obtain normal network encrypted traffic from CICIDS-2017 to complete the processing of the data set.

(2) Deep learning algorithms such as 1D-CNN, 2D-CNN, CNN-LSTM and LSTM are trained on the dataset. LSTM is even better in this training set, with an accuracy rate of 99.84% and a false alarm rate as low as 0.80%., performs well in CPU and memory usage.

## 2 RELATED WORK

In the encrypted traffic detection test based on TLS protocol, even if the same encryption mechanism is used, the encrypted traffic will show different data distribution characteristics due to the different distri-bution and utilization of the original traffic. Therefore, most researchers use binary classification of encrypted traffic, that is, to identify malicious traffic among legitimate traffic. Currently, flow-based machine learning and deep learning methods are the mainstream methods for encrypted traffic classification [6]. Shekhawat et al. [3] proposed three machine learning techniques, Random Forest (RF), Support Vector Machine (SVM) and XGBoost to distinguish malicious encrypted traffic from benign encrypted traffic. The learned model acquires traffic features. Stegiopoulos et al. [4] used seven different supervised learning algorithms such as K-Nearest Neighbors (KNN), Regression Trees (CART) and Naïve Bayes for comparison Experiment to detect malicious traffic from datasets with multiple encryption protocols. However, traditional machine learning algorithms suffer from several shortcomings in encrypted traffic identification. 1) IoT encrypted traffic identification and classification Encryption protocols are usually not unique in the entire network environment, and other protocols have similar characteristics, which significantly degrades the performance of machine learning to identify encryption protocols in the backbone network. 2) The establishment time of the detection model is largely determined by the machine learning algorithm. There are more than 100 features available, and more than a dozen machine learning algorithms [1]. Determining the most salient features of each cryptographic protocol in practical use and designing models with appropriate machine learning algorithms requires considerable effort. 3) Machine learning models are time-consuming to extract features from encrypted traffic. Due to the complexity of the algorithm, it may even be slower and therefore less efficient than manual feature labeling [7].

## 3 INTRUSION DETECTION SYSTEM DESIGN

### *3.1 System Design*

The intrusion detection system proposed in this paper is designed based on the network layer of the Internet of Things. The architecture of the Internet of Things can be divided into three layers, from top to bottom, they are the application layer, the network (transmission) layer and the perception (edge) layer [2]. Among them, applications and service functions run in the application layer, and endpoints and terminal devices collect environmental information and run in the perception layer.

In the operation of the Internet of Things, the perception layer transmits the acquired information to the gateway or server, and the gateway or server provides services or interacts externally through the Internet. In this paper, the intrusion detection system is deployed at the network layer based on three considerations: 1) There are a large number of devices at the perception layer, and the value of a single device being at-tacked is low. Considering the deployment cost of the intrusion detection system, it is not suitable for deployment on a large number of devices at the perception layer. 2) There are many protocols deployed in the perception layer, and it is difficult to extract the encrypted protocol information to train the detection model. 3) The application layer is the external service layer, with high level and large business volume, which is not suitable for running the intrusion detection system from time to time. Therefore, this paper chooses to deploy the intrusion detection system at the network layer, and uses the traffic generated by the TLS encryption protocol for training.

### *3.2 Dataset Processing*

Traffic data for 20 different types of malware categories, labeled "malicious traffic", was extracted from the dataset CTU-13 provided by the Cybersecurity Group at the Czech Technical University in Prague. And the normal traffic data is extracted from the CICIDS-2017 dataset, marked as "normal traffic". Two datasets are selected to extract normal encrypted traffic and abnormal encrypted traffic respectively to increase the generalization of the dataset, thereby creating a new set of network traffic datasets to meet the labeling requirements of binary classification. The constructed dataset is fed into the traffic feature extractor Cisco Joy for extracting more than 200 traffic features. The flow

feature extractor extracts the original data file and combines long network flows from inbound and outbound directions into data records with a length of up to 200 data packets, and network flows with less than 200 data packets remain unchanged. De-identify data by masking source and destination IP addresses. Doing so removes the bias introduced by the static IP address used to capture the network stream host. Then, add the "id" and "Label" for each stream, after which the file is output in JSON format. Finally, to ensure that the datasets are all encrypted, we filter the TLS-encrypted stream by checking if there are TLS-encrypted features available in the stream. Hence, 114000 encrypted streams are obtained. The model input of deep learning is often input in the form of a two-dimensional matrix, with a row of data representing a stream instance and a column of data representing a class of eigenvalues. Metadata features include statistical information features that can be calculated for each flow, including source port, destination port, number of inbound packets, number of outbound packets and flow duration, etc. Metadata features extracted for each flow. At the same time, TLS protocol characteristics not only include numerical characteristics, such as the number of cipher suites provided by the client and the server, the number of TLS encrypted packets contained in the stream, and some other variable-sized arrays of numerical characteristics, such as the number of cipher suites provided by the client and the server. Cipher suites, TLS extensions supported by the client, etc. Therefore, these variable-sized array features must be converted to fixed-length numerical features, so that TLS protocol features and metadata features can be used in the input data matrix. Process data according to the TLS signature of the malware family. When training a deep learning classifier, the selection of important but small number of features plays a crucial role. Therefore, a feature selection method called chi-square test is used [5], which has also been widely used in many previous studies, and has been used in the fields of network traffic analysis and intrusion detection systems. accomplish. Thus we can rank the features according to their correlation with the target variable, and can provide the proposed classification model with a subset of features that are highly correlated with the classification labels. By chi-square test, we set to filter out the top N=50 features.

### *3.3 Deep Learning Algorithm Model And Model Training*

This paper, the 1D_CNN, 2D_CNN, LSTM and CNN_LSTM models are implemented, and the ex-tracted flow features are provided as input data to the CNN and LSTM models respectively, and each parameter of the neural network is trained through a large number of labeled flow data, and then get bet-ter-performing classifier. For the 2D_CNN model, this paper reshapes the input data from 50×1 to 10×5. The model architectures of its one-dimensional CNN and two-dimensional CNN are shown in Figure 1 and Figure 2.

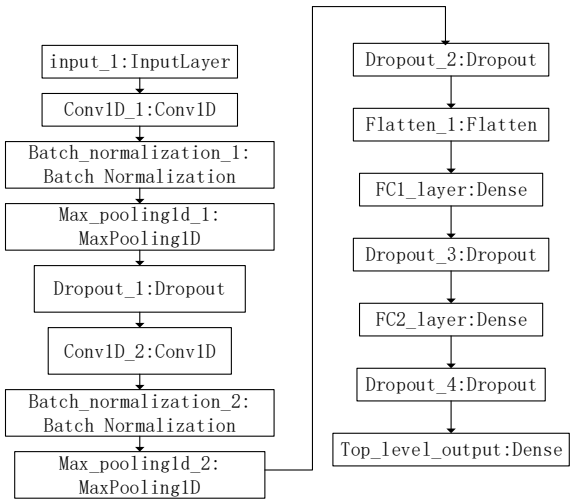


**Figure 1** Structure diagram of one-dimensional CNN convolutional neural network

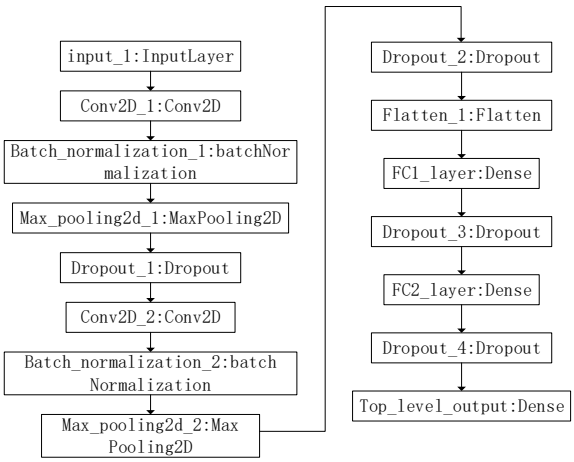


**Figure 2 S**tructure diagram of two-dimensional CNN convolutional neural network

For the LSTM long-short-term memory network model, the LSTM layer uses temporal information to encode the representation of the given sequence data, so the LSTM layer requires two-dimensional input data with a size of n×D, in this paper n=10, D=5. The LSTM classifier constructed in this paper contains a total of two LSTM layers and an output softmax layer to form a recurrent neural network (RNN), as shown in Figure 3.

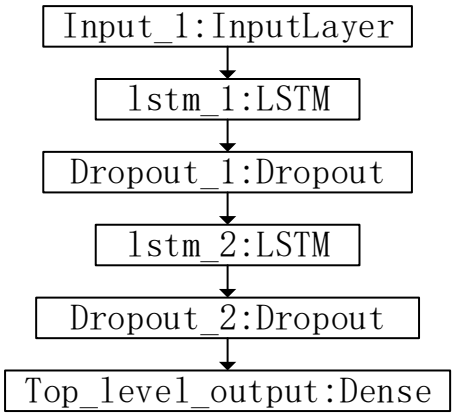


**Figure 3** LSTM long and short-term memory network structure diagram

Since the CNN convolutional neural network is good at extracting the spatial features of the traffic, the LSTM long and short-term memory network is good at extracting the time series features of the traffic and capturing the possible dependency information before and after the traffic. Therefore, it is considered to integrate the CNN convolutional neural network with the LSTM long and short-term memory network, so as to extract as many spatiotemporal correlation features among many data packets in the network flow as possible and improve the performance of the classifier. We use CNN layers to exploit spatial information in the early stages of model training and LSTM layers in the later stages of model training, the CNN_LSTM hybrid neural network is shown in Figure 4.

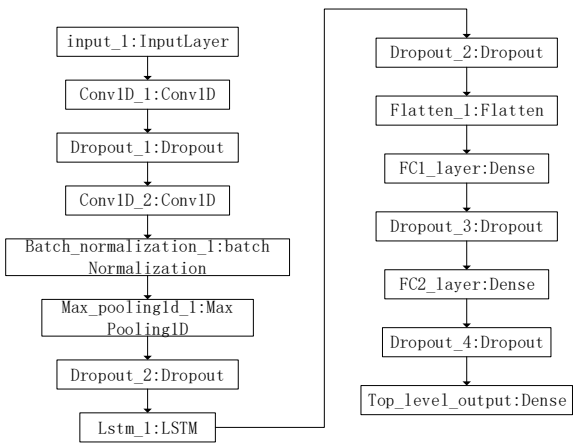


**Figure 4** CNN_LSTM hybrid neural network structure diagram

## 4 EXPERIMENTAL DESIGN AND ANALYSIS

### *4.1 Experimental Design*

To evaluate the proposed method, this paper uses the Keras library behind Tensorflow for deep learning models on GPUs. The hardware configuration environment is Windows 10 Professional Edition operating system, Intel(R) Xeon(R) Bronze 3206R CPU @1.90GHz processor, NVDIA Quadro P2000 5G+15.8G GPU, 32.0GB onboard RAM. After many parameter adjustments and experiments, the network parameters of this paper are determined:

1 D_CNN: the size of the convolution kernel is 3×3, the stride is 1, the convolution layer contains 128 convolution kernels, the fully connected layer contains 128 neurons, the activation function is the "relu" function, and the deactivation The dropout rate is 0.5.

2D_CNN: Same as 1D_CNN.

LSTM: The number of LSTM layers is set to 2, the number of hidden layer neurons is 128, and the dropout rate is 0.5.

CNN_LSTM: The size of the convolution kernel is 5×1, the stride is 1, the convolution layer contains 32 convolution kernels, the number of neurons in the hidden layer is set to 200, and the inactivation rate is set to 0.5.

Also, for all the above deep learning models, hyperparameters are optimized by manually searching for default values, such as learning rate set to 0.001, decay rate of learning rate set to 0.0001, batch size set to 0.0001 is 100, and the regularization parameter of the output layer is set to 0.0001 to prevent overfitting. All models were trained for 10 epochs using the Adam optimizer.

### *4.2 Data Analysis*

This paper selects parameters such as accuracy rate (Accuracy, Acc), precision rate (Precision, P), recall rate (Recall, R), false alarm rate (False Alarm Rate, FAR) and F 1-Measure (F 1) Evaluate the model. The calculation formula of each parameter is as follows:

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

$$P = \frac{TP}{TP + FP} \tag{2}$$

$$R = \frac{TP}{TP + FN} \tag{3}$$

$$F_1 = 2\frac{P \bullet R}{P + R} \tag{4}$$

$$FAR = 1 - P \tag{5}$$

Among them, the meanings of parameters such as TP, FP, TN, and FN are shown in Table 1.

**Table 1** Parameter meaning

| Evaluation indicators | Parameter representation |
|---|---|
| TP (true positives) | The number of normal traffic that is predicted to be normal |
| FP (false negatives) | The number of malicious traffic that is predicted to be normal |
| TN (true negatives) | The number of malicious traffic that is predicted to be malicious |
| FN (false negatives) | The number of normal traffic that is predicted to be malicious |

After 4 deep learning model classifiers such as 1D_CNN, 2D_CNN, LSTM and CNN_LSTM are trained on the dataset, the evaluation parameters Acc, P, R, FAR, F1 of different model classifiers are shown in Table 2

**Table 2** The performance of each deep learning classifier

| Model | Acc | P | R | FAR | F1 _ |
|---|---|---|---|---|---|
| 1D CNN | 99.05% | 9 8.65% | 99.12% | 1.35% | 0.9888 |
| 2D CNN | 99.44% | 9 8.01% | 99.69% | 1.99% | 0.9884 |
| LSTM | 99.84% | 9 9.20% | 99.96% | 0.80% | 0.9958 |
| CNN-LSTM | 99.45% | 97.91% | 99.72% | 2.09% | 0.9881 |

Accuracy is a direct indicator of model performance evaluation. Comparing 4 deep learning models, LSTM has the highest classification Acc of 99.84%, but a single indicator cannot characterize the quality of the model. Horizontal comparison of P, R, FAR and F1 4 indicators. Among the indicators of the intrusion detection model, the level of R is a measure of the ability of the model to check all malicious traffic. LSTM has the highest R among the four algorithms, reaching 99.96%, and has the strongest ability. At the same time, the performance of the LSTM model on P is the best, reaching 99.20%, that is, the minimum false alarm rate is 0.8 %, which meets the requirements of a lower false alarm rate for the security system. The performance of the four deep learning algorithms in the comprehensive evaluation index F1, LSTM is also the best, reaching 0.9958. Therefore, LSTM performs best on the 5 model evaluation metrics Acc, P, R, FAR, and F1.

### *4.3 Energy Analysis*

A large number of IoT devices are connected to the network, and the amount of data is huge. Therefore, the resource requirements of gateways or servers directly connected to the Internet are relatively high. Therefore, when deploying an intrusion detection system, the consumption of device resources should be fully considered when the system is deployed. In order to detect the energy consumption occupied by the deployment of the intrusion detection system, all deep learning models were built using the python language and deployed with TensorFlow-gpu version 1.6.0. On the basis of the same hardware, common deep learning algorithms are tested, and the CPU ratio and memory usage of the intrusion detection system are detected by calling Python 's psutil and memory_profile databases. The energy consumption analysis of each deep learning model is shown in Table 3.

**Table 3** Energy analysis of deep learning models

| Model | CPU (%) | Memory (MB) |
|---|---|---|
| 1D - CNN | 10.836 | 602.90 |
| 2D-CNN | 5.770 | 815.08 |
| LSTM | 9.445 | 721.11 |
| CNN-LSTM | 8.323 | 772.78 |

3 that the CPU occupancy of the LSTM model is 9.445 %, which is slightly higher than 5.77 % and 8.323 % of 2D_CNN and CNN_LSTM, and lower than 10.836 % of 1D_CNN. In terms of memory footprint, the memory required by LSTM is 721.11MB, which is slightly higher than that of 1D_CNN and lower than that of 2D-CNN and CNN-LSTM. LSTM is comparable to other deep learning models in terms of resource consumption such as memory and CPU.

## 5 CONCLUSIONS

This paper describes the problems of time-consuming feature extraction and general efficiency in the traditional machine learning method for detecting malicious encrypted traffic. The deep learning algorithm LSTM is used to construct the intrusion detection system. The malware traffic data is extracted from the CTU-13 dataset and the normal traffic data is extracted from the CICIDS-2017 dataset to generate a binary classification encrypted traffic dataset to train the LSTM intrusion detection model. In order to prove the better performance of the LSTM model in this dataset, this paper trains three deep

learning models, 1D_CNN, 2D_CNN and CNN -LSTM at the same time, through the precision, accuracy, false alarm, recall and F1 The comparison of five model evaluation indicators confirms that the performance of the LSTM model is better, and the accuracy of the intuitive classification performance indicators is as high as 99.84%. At the same time, LSTM has a similar level with the other three models in terms of memory and CPU usage. Therefore, the ability of long short-term memory network (LSTM) in malicious traffic detection meets the needs of IoT intrusion detection.

## REFERENCES

[1] Duan M. (2018) A Parallel Multiclassification Algorithm for Big Data Using an Extreme Learning Machine [J]. IEEE Transactions on Neural Networks and Learning Systems,29(6):2337-2351.

[2] Li Y. (2021) A Survey of Encrypted Malicious Traffic Detection[C]//, IEEE, 2021.

[3] Shekhawat A S. (2019) Feature analysis of encrypted malicious traffic[J]. Expert systems with applications,125:130-141.

[4] Stergiopoulos G. (2018) Automatic Detection of Various Malicious Traffic Using Side Channel Features on TCP Packets[M]//Cham: Springer International Publishing:346-362.

[5] Shen C. (2021) The Chi-Square Test of Distance Correlation [J]. Journal of Computational and Graphical Statistics:1-21.

[6] Wang Z. (2022) Machine learning for encrypted malicious traffic detection: Approaches, datasets and comparative study[J]. Computers & Security, 113:102542.

[7] Zhao Y. (2021) Edge Intelligence Based Identification and Classification of Encrypted Traffic of Internet of Things[J]. IEEE Access, 9: 21895-21903.