ISSN: 2688-8645

IoT-based Hybrid Deep Learning Intrusion Detection System

Ning Dong, Xiaorong Cheng

School of North China Electric Power University, Baoding, 071000, China

Abstract

With the increasing number of Internet of Things(IoT) access devices around the world, and the safety management awareness of operation and maintenance personnel for IoT devices is not as good as that of traditional servers and PC, a large number of IoT devices have been compromised by hackers and become part of botnets. Mines, DDoS and other means seek black profits for hackers. Based on the intrusion detection dataset for IoT devices and botnets, this paper designs a hybrid deep learning intrusion detection model based on convolutional neural network (CNN) and long short-term memory (LSTM) to detect occurrences in the IoT environment various suspicious attack traffic. Among them, CNN is used to extract meaningful features in the dataset, and because the weight sharing reduces the running time, LSTM performs further processing and judgment on the features extracted by CNN. After experimental analysis, the model has reached an accuracy rate of more than 95%, and for most label, the accuracy rate, precision rate, recall rate and other metrics have reached more than 95%.

Keywords

Intrusion detection; IoT; Deep learning; CNN; LSTM.

1. Introduction

With the continuous development and improvement of wireless communication technology, the distance of wireless communication is getting farther and farther, the communication quality is getting higher and higher, and the energy consumption is getting lower and lower. More IoT devices are beginning to connect to the Internet. As of the end of 2020, a total of 50 billion IoT devices have been connected to the Internet [1], and the global IoT security expenditure has reached 3.1 billion dollars [2]. As the scale of the IoT continues to increase, The attack method has gradually shifted to the IoT. At the same time, due to the interconnected nature of the IoT, hackers' attacks on the IoT will cause more serious harm. For example, the Stuxnet virus attack against real-world facilities that first appeared in 2010, and Mirai first appeared in 2016. It acquired a large number of IoT device permissions by scanning open telnet ports, weak password blasting, etc., and formed a large-scale botnet and use botnet to launch DDoS attacks on dns service providers, which made dozens of platforms such as Twitter and Amazon inaccessible.

Intrusion detection system (IDS) is a network security device that can track and audit the traffic and user behavior in the network in real time, determine whether the current operation is suspicious, and actively alarm. Different from other network security equipment, IDS is an active defense system. According to the source of information, intrusion detection can be divided into host-based and network-based. According to the detection method, it can be divided into abnormal intrusion detection type and misuse intrusion detection type. Some early IDS established a user feature table, compared the current feature with the previously stored and finalized features, and used probability statistical methods to determine whether it was an attack. However, the disadvantage of this method is that the user feature table needs to be manually updated, and it is difficult to determine the parameters for determining whether intrusion or not, which easily causes an excessively high false alarm rate and a false alarm rate.

ISSN: 2688-8645

With the development of machine learning technology, more and more papers apply machine learning and deep learning technology to the realization of IDS, and have achieved considerable results. However, most of the papers currently appearing use dataset and intrusion detection methods for host and network, rather than attack detection for IoT devices, resulting in relatively few IDS for IoT devices.

Based on the intrusion detection dataset collected in the IoT environment, this paper designs a CNN+LSTM model to realize an IDS for the IoT environment. The system shows relatively high performance and accuracy.

The content of this article is divided into the following five parts:

1. The current security status of the IoT environment and the introduction of IDS.
2. Currently known intrusion methods and IDS for the IoT environment.
3. Introduction to the model architecture and dataset of the IDS based on CNN-LSTM proposed in this paper.
4. Implementation and evaluation of the proposed system model.
5. Conclusions and future work.

2. Relate Work

Due to the small size of the IoT devices, and some devices are powered by batteries, the storage and computing capabilities of the IoT devices are weaker than traditional servers and personal PC. The current attack methods for the IoT are mainly as follows:

1. Vulnerability scanning: Most of the software inside the IoT is embedded, and some software is difficult to update, because there are more devices with vulnerabilities in the IoT environment. Vulnerability scanning of the IoT devices is a relatively common type of information detection and attack means.

2. Weak password brute force: Due to the large number of IoT devices, and some administrators are not sufficiently aware of the security of the IoT environment, a large number of networks are exposed to the public network, and the account passwords are the default devices when they leave the factory. Through these devices, hackers can control IoT devices, and then enter the internal network of the victim's organization, causing greater damage.

3. Flooding: Although the computing power of IoT devices is lower than that of traditional PC devices, due to the large number of IoT devices in the current network environment, if you can control multiple IoT devices to launch a DDoS attack on a victim, it can still cause serious consequence.

4. Botnet: As there are so many IoT devices that hackers have access to through vulnerability scanning, weak passwords, etc., which cannot be managed one by one, there is a botnet, that is, there are one or more bot herder , and hackers control their access through the bot herder. Through this method, hackers can use the obtained IoT devices to scan other devices, brute force weak passwords to expand the scale of their botnets. At the same time, use the acquired IoT devices for flooding, mining and other operations for profit.

Muder Almiania [3] proposed a multi-layer RNN model based on fog computing, using the NSL-KDD dataset to achieve intrusion detection based on the IoT, and adding two new matrices to have a deeper understanding of the model. Then the security of the IoT fog computing node itself is difficult to guarantee, and once it breaks through the IDS will also fail.

Danish Vasan [4] and others proposed to use the RNN+CNN model to detect the security of the software in the IoT devices. Determine whether the device is invaded by checking whether there is malware in the device. This method realizes the detection of cross-platform software and achieves high accuracy. However, it cannot detect attacks from the network, such as DDoS attacks launched by other IoT devices on the victim.

ISSN: 2688-8645

KISHWAR SADAF [5] and others proposed using Auto-Encoder and Isolation Forest to detect the traffic in the network in real time under the fog computing platform. The paper achieved a high accuracy rate in the case of two classifications. However, the two-category is not well applied in practical applications, and network maintenance personnel cannot deploy corresponding emergency response measures based on the two-category results.

Fal Sadikin [6] proposed an anomaly-based IDS that combines rules and machine learning under the ZigBee platform, where rules are used to detect known attack methods, and machine learning is used to detect unknown intrusion methods.

3. Dataset and Model

3.1. Data Precessed

The dataset [7] used in this article is a botnet dataset made by Google for the IoT environment. The dataset has 620,000 pieces of data and 83 features, including some flow-based features, so that researchers can design an IoT IDS based on flow information. There are three labels in the dataset, including a binary-class label and two multi-category labels. One of the multi-category labels includes the flow of traffic divided into normal flows and basic attack types, and the other label subdivides the basic attack types to classify more attack types. This article uses the first multi-category label with four attack types: Scann, DDoS, ARP man-in-the-middle(MITM) spoofing, and Mirai. Among them, Mirai is a botnet that appeared around 2016. The data distribution of this datset is shown in Table 1:

Table 1. Dataset label distribution

Label	Normal	Mirai	DoS	ARP	SCAN
Num	40073	415677	59391	35377	75266

In the process of dataset processing, due to the large value difference between some features, the features are normalized. The normalization formula is as follows:

$$X_{normal} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

In the normalization process, due to the large gap between the features of the data, some data appear to be Nan after the feature is normalized, which affects the subsequent training process. Therefore, these features are deleted. At the same time, some features, such as source IP, destination IP, timestamp and other features only have relevance in the dataset, and cannot be applied in reality, so this part of the feature are also deleted.

The data is divided into training set and test set by 7:3. At the same time, the label in the dataset are processed by one-hot encoding. And the training set and test set are both trained and tested with 32 samples as a batch.

3.2. Model Architecture

The model in this paper uses the CNN+LSTM architecture. Among them, CNN is used to extract spatial features of data, and LSTM is used to extract temporally features. At the same time, because CNN is a shared convolution kernel, it greatly reduces the amount of parameters and shortens the running time. To a certain extent, it meets the low computing power and real-time requirements in the IoT environment. The model architecture is shown in Figure 1:

ISSN: 2688-8645

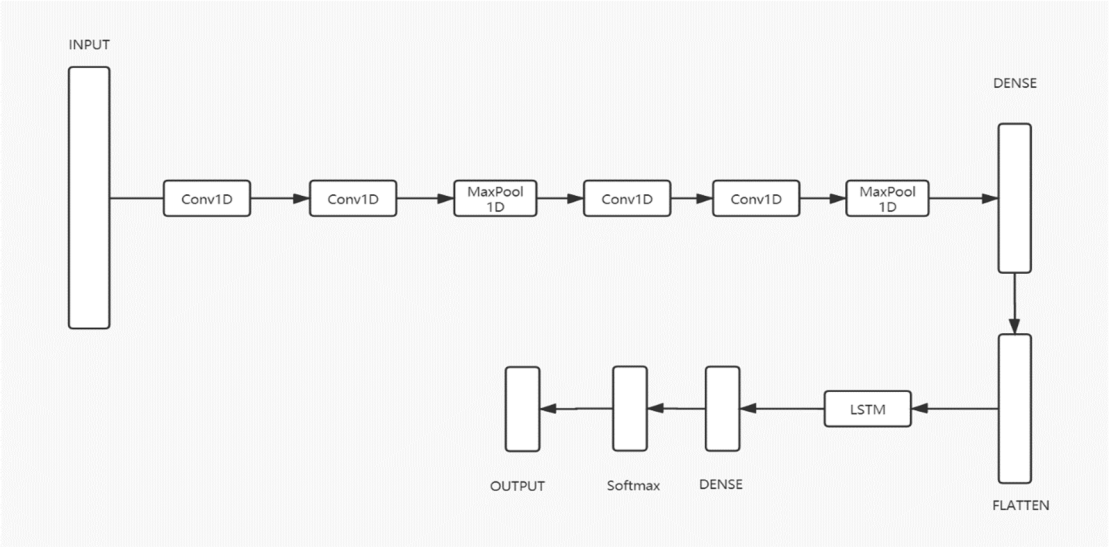

Figure 1. Model architecture

As shown in Figure 1, the model is divided into a convolution part and an LSTM part. In order to ensure the accuracy of the model while trying to avoid too few features, the convolution part performs a maximum pooling layer after every two layers of Conv1D operation. At the same time, the dropout operation is performed after pooling to ensure the sparsity of the matrix and avoid overfitting. After the convolution is over, the output of the convolutional layer is stretched into a one-dimensional vector and enters the LSTM network. Finally, the Softmax classifier is added, and the model ends and outputs.

3.2.1. CNN

CNN is a special deep neural network model. Its particularity is reflected in two aspects. On the one hand, the connections of its neurons are not fully connected, and on the other hand, the weights of connections between certain neurons in the same layer are shared. Its non-fully connected and weight-sharing network structure makes it more similar to a biological neural network, reducing the complexity of the network model (for deep structures that are difficult to learn, this is very important), and reducing the weight quantity. Compared with the traditional deep neural network, the advantages of CNN are as follows:

1. Weight sharing: In the convolutional neural network, each convolution filter of the convolution layer repeatedly acts on the entire receptive field, convolving the input data, and the convolution result constitutes the feature map of the input data, which is extracted the local characteristics of the data. Each convolution filter shares the same parameters, including the same weight matrix and bias terms. The advantage of sharing weights is that the location of local features does not need to be considered when extracting features from the data. Moreover, weight sharing provides an effective way to greatly reduce the number of convolutional neural network model parameters to be learned.

2. Maximum pool sampling: a non-linear down-sampling method. After the data features are obtained by convolution, these features are used for classification. All the extracted feature data can be used to train the classifier, but this usually results in a huge amount of calculation. Therefore, after acquiring the convolutional features of the data, the dimensionality of the convolutional features should be reduced by the maximum pool sampling method. The convolution feature is divided into several n*n disjoint regions, and the maximum (or average) feature of these regions is used to represent the reduced dimensionality convolution feature. These dimensionality-reduced features are easier to classify.

3. Sparse connection: In the BP neural network, the neuron nodes of each layer are a linear one-dimensional arrangement structure, and each neuron node of the layer is fully connected. In

ISSN: 2688-8645

convolutional neural networks, the neuron nodes between layers are no longer fully connected, and the local spatial correlation between layers is used to connect the neuron nodes of each adjacent layer to only the upper neuron nodes that are close to it. This greatly reduces the parameter scale of the neural network architecture.

3.2.2. LSTM

LSTM is a network proposed on the basis of RNN in order to solve the problem that RNN cannot handle long-term dependent information. The LSTM network structure is shown in Figure 2.

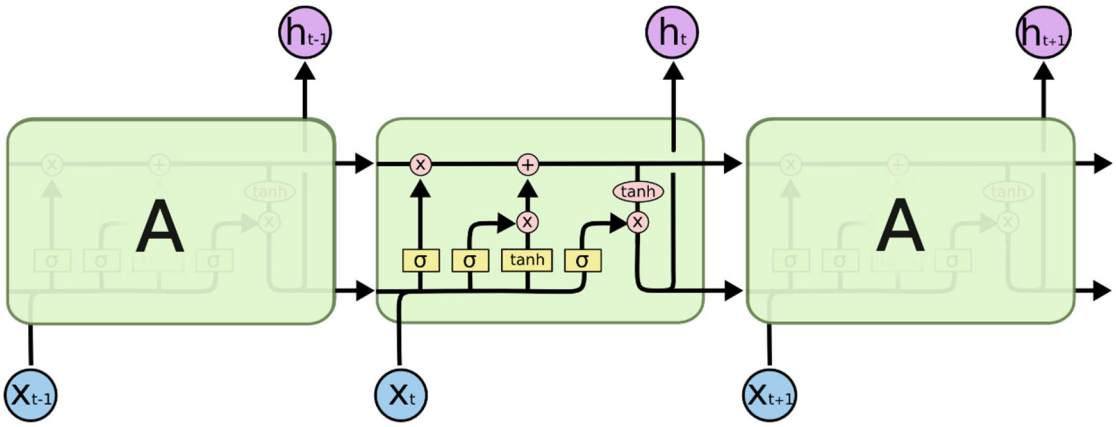

Figure 2. LSTM architecture

Both LSTM and RNN are composed of cell links with the same structure, and each cell is divided into three gates and one cell state. The update of the cell state is determined by three gates. The three gates are the forget gate, the input gate, and the output gate. The forget gate determines to retain part of the data from the previous cell, the input gate determines to retain part of the current input information, and the output gate determines the value of the next hidden state. The corresponding formulas for the three gates are as follows:

$$\text{Forget gate: } f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2}$$

$$\text{Input gate: } i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3}$$

$$\widetilde{C_t} = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{4}$$

$$\text{Output gate: } o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * \tanh(C_t) \tag{6}$$

Cell state update formulas as follows:

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t}. \tag{7}$$

Where W is the parameter of the corresponding three gates, b is the corresponding bias term, h_t represents the value of the hidden state at time t, and x_t represents the input value at time t. σ is the activation function of the three gates, generally is sigmoid function. C_t is the state of each cell in the LSTM network. Through the update of the three gate states, LSTM can retain the

ISSN: 2688-8645

information of the data at the previous one or several moments in the data, thereby solving the problem of long-term data dependence in the RNN network.

4. Experiments and Evaluation

4.1. Experiments Design

This experiment's dataset uses IoT Intrusion Dataset 2020, which is produced and made public by Google. This dataset is a flow-based IoT intrusion detection dataset focusing on botnets. A total of 620,000+ pieces of data, there are four basic types of attacks and 83 characteristics. Taking into account the practical application and feature normalization, 7 features were deleted.

In the process of implementing the experimental model, after continuous experiments and try, the network structure and parameters are finally determined. Among them, the convolutional neural network part has 32 convolution kernels of size 3 in each layer, the activation function is relu, and the padding mode is same. The maximum pooling layer size is 2, and the step size is 2. The dropout layer parameter is 0.25. After the convolution part is finished, it enters the LSTM network through a fully connected layer with 32 neurons. The LSTM network outputs 32 neurons, and finally enters the softmax classifier and outputs. The cost function of the model uses the cross-entropy cost function, the Adam optimizer is used, and the learning rate is 0.0001. After 30 epoch, the overall accuracy of the model has reached more than 95%.

4.2. Evaluation and Analysis

Table 2 shows the results of the model's judgment on 18,000 test sets. It can be seen that the samples have serious imbalances, and Mirai accounts for the vast majority of the samples. The evaluation of the model is finally measured by four metrics, namely accuracy rate, precision rate, recall rate and F1-SCORE. The calculation formulas of the four metrics are as follows:

Table 2. Performance of the model on the test set

Predict / Label	DoS	ARP	Mirai	Normal	Scan	Count
DoS	1681	0	2	5	0	1688
ARP	0	805	195	1	0	1001
Mirai	0	149	11608	5	213	11975
Normal	2	4	23	1116	0	1145
Scan	0	23	88	0	2080	2191
Count	1683	981	11916	1127	2293	18000

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{8}$$

$$Precision = \frac{TP}{TP+FP} \tag{9}$$

$$Recall = \frac{TP}{TP+FN} \tag{10}$$

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precison+Recal} \tag{11}$$

ISSN: 2688-8645

TP represents the number of samples that determine the attack as an attack, TN represents the number of samples that determine the normal type as a normal type, FP represents the number of samples that determine the attack as a normal type, and FN represents the number of samples that determine the normal type as an attack.

Table 3 shows the numerical value of each mertrics corresponding to each label. It can be seen that all metrics of each tag except ARP attacks maintain a value of more than 90%, and in the four attack types, DoS and Normal can still maintain a relatively high accuracy rate, precision rate and recall even when the number of samples is sparse. The ARP attack is difficult to distinguish because it is located between the second and third layers of the TCP/IP protocol. The determination of the label is also one of the problems that need to be solved later.

Table 3. Corresponding metrics of each label

Lable / Metrics	DoS	ARP	Mirai	Normal	Scan
Accuracy	0.9995	0.97933	0.9575	0.99788	0.982
Precision	0.995853	0.804196	0.969353	0.974672	0.949338
Recall	0.998812	0.820591	0.974152	0.99024	0.907109
F1-SCORE	0.498665	0.406155	0.485873	0.491197	0.463872

Figure 3 shows the model's training set and test set accuracy and loss function change curve during the training process. In actual training, due to the Dropout layer, the accuracy and loss of the model in the test set are better than the training set.

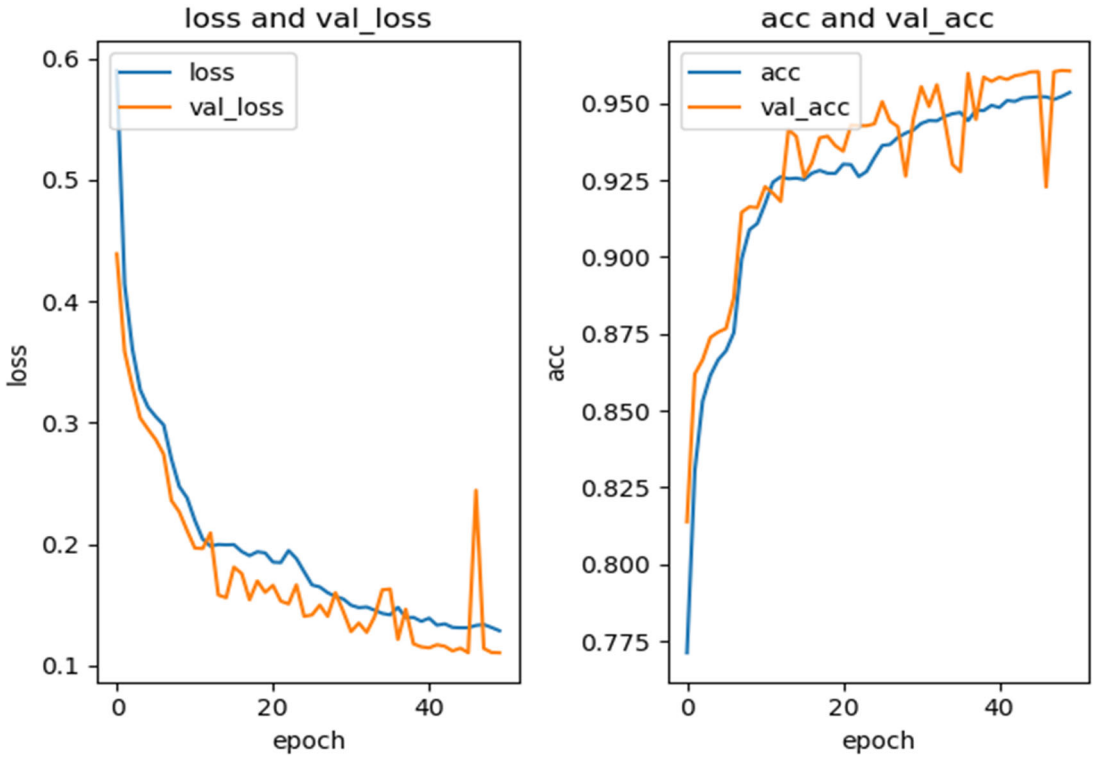

Figure 2. Metrics curve during training

ISSN: 2688-8645

5. Conclusions

The paper proposed an IDS based on flow characteristics, collected dataset under the IoT equipment for botnets, and trained using the CNN+LSTM network architecture. The system uses CNN to extract influential features from the data and uses LSTM to make judgments. In the end, after constant attempts and adjustments, the accuracy rate and other metrics are higher than 90%, and most of the models with label accuracy higher than 95%. Due to the imbalance distribution of samples and the special types of attacks, the accuracy of some tags is slightly lower. How to solve this problem is the direction of follow-up research. At the same time, due to the huge IoT environment and numerous devices, how to complete the identification as quickly as possible while ensuring accuracy is also one of the tasks that this paper needs to do in the future.

References

[1] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali and M. Guizani, "A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security," in IEEE Communications Surveys & Tutorials, vol. 22, no. 3, pp. 1646-1685.

[2] G. W. Cassales, H. Senger, E. R. de Faria and A. Bifet, "IDSA-IoT: An Intrusion Detection System Architecture for IoT Networks," 2019 IEEE Symposium on Computers and Communications (ISCC), Barcelona, Spain, 2019, pp. 1-7.

[3] Muder Almiani, Alia AbuGhazleh, Amer Al-Rahayfeh, Saleh Atiewi, Abdul Razaque,Deep recurrent neural network for IoT intrusion detection system,Simulation Modelling Practice and Theory,Volume 101,2020,102031,ISSN 1569-190X.

[4] D. Vasan, M. Alazab, S. Venkatraman, J. Akram and Z. Qin, "MTHAEL: Cross-Architecture IoT Malware Detection Based on Neural Network Advanced Ensemble Learning," in IEEE Transactions on Computers, vol. 69, no. 11, pp. 1654-1667.

[5] K. Sadaf and J. Sultana, "Intrusion Detection Based on Autoencoder and Isolation Forest in Fog Computing," in IEEE Access, vol. 8, pp. 167059-167068.

[6] Sadikin, F. and Kumar, S. (2020). ZigBee IoT Intrusion Detection System: A Hybrid Approach with Rule-based and Machine Learning Anomaly Detection.In Proceedings of the 5th International Conference on Internet of Things, Big Data and Security - Volume 1: IoTBDS, ISBN 978-989-758-426-8, pages 57-68.

[7] Ullah I., Mahmoud Q.H. (2020) A Scheme for Generating a Dataset for Anomalous Activity Detection in IoT Networks. In: Goutte C., Zhu X. (eds) Advances in Artificial Intelligence. Canadian AI 2020. Lecture Notes in Computer Science, vol 12109.

[8] Information on http://colah.github.io/posts/2015-08-Understanding-LSTMs