Network Traffic Anomaly Detection Method Based on CAE and LSTM

Shuai Guo, Yao Liu and Yang Su*

College of Cryptographic Engineering, Engineering University of PAP, Xi’an, Shanxi, China
Email: 1028618940@qq.com; liuyaoly61@foxmail.com; *suyang75@126.com

Abstract. This paper constructs a deep learning method for detecting network traffic anomalies to enhance the secure transmission of data in networks due to the complex, diverse and numerous types of anomalous traffic in current networks. The method combines multiple convolutional auto-encoders (Multi-CAE) with a long short-term memory network. The convolutional auto-encoders are obtained by combining stacked auto-encoders with convolutional layers, which can not only reduce feature loss but also effectively extract the spatial structure of samples. The use of Multi-CAE greatly improves the feature extraction capability, and combined with the long short-term memory network to extract temporal features, the effective features extracted in this paper are more comprehensive and less losses compared to the models used in other researches. A comparison of the loss values in the training of CAE (Convolutional Auto-Encoders) and SAE (Stacked Auto-Encoders) in the experiments shows that the loss values of CAE are about one-tenth lower than those of SAE, and the method consisting of Multi-CAE and LSTM for the USTC-TFC2016 dataset was trained with accuracy values up to 99.98%, and the precision, recall and f1-score parameters were also above 99%, outperforming other studies.

Keywords. Deep learning; convolutional auto-encoder; spatial structure; temporal features; anomaly detection.

1. Introduction
With the currently rapid development of cloud computing, big data, 5G networks, artificial intelligence, blockchain and other technologies, the number of problems related to abnormal traffic in the network is increasing, and the damage caused is becoming more and more serious every year. To reduce the occurrence of these problems as much as possible and protect the security of all kinds of information in the network, it is necessary to strengthen the detection and supervision of all kinds of traffic in the network, especially the abnormal data found in the transmission process, which requires more focused attention.

Although there has been a lot of research into anomaly detection related methods [1], such as using supervised and unsupervised learning for detection, or using traditional machine learning or deep learning methods for detection, the results are also relatively good. However, in real life situations the problem becomes very complex and the requirements for detection are high, which is difficult to achieve using only a single model for detection. It has been proven that combining different models with each other can achieve higher efficiency and better results. In this paper, we propose a method that automatically unfolds network traffic anomaly detection. The method can make use of a combination of convolutional auto-encoder networks and long short-term memory networks to extract...
spatial and temporal features in data traffic, which improves the processing efficiency and detection effectiveness to a certain extent. The main contributions of this paper are as follows:

(1) A Multi-CAE and LSTM based anomaly traffic detection model was constructed. The anomaly detection of 20 types of network traffic was achieved with an accuracy of 99.98%, and the precision, recall and f1-score parameters were also above 99%.

(2) In order to compare the effect of CAE and SAE models, a set of experiments on the comparison of loss values is set up in this paper. By training the two models with decreasing loss values, the results show that CAE is significantly lower than SAE, proving that CAE outperforms SAE.

2. Related Work
The concept of anomaly was introduced by Chandola [2] in 2009: anomalies are patterns in data that do not conform to a well defined notion of normal behavior. The anomaly flow is relatively to normal flow, such as anything not normal is anomalous. Anomaly traffic often varies with time and environment. For example, the encrypted traffic of the SSL protocol that is currently in use was anomalous traffic when it first appeared, but over time, as it became more widely used and more widely accepted by the public, it became normal network traffic. The three TCP handshakes were normal interaction messages, but when used in a flooding attack they also became anomalous data. Anomalous traffic is therefore relative and must be analysed in relation to normal data. The appearance of anomalous data in practical applications often represents a problem and a potential hazard, and therefore it is necessary to study it. Among the existing methods for studying anomalies the supervised learning approach is the commonly used, quickly and easily and with a high degree of accuracy. Some of the studies are:

References [3, 4] introduced some deep learning methods in cyber security applications, including Restricted Boltzmann Machine, Auto-encoders, Generative Adversarial Networks, Convolutional Neural Networks and Recurrent Neural Network, each of these models has its own characteristics, such as AE, which is an unsupervised neural network that can change dimensions in the hidden layer to create a higher or lower dimensional representation of the data. GANs contain a generator and a discriminator, with the generator receiving input data and generating output data with the same characteristics as the real data, and the discriminator receiving real data and data from the generator and trying to distinguish whether the input is real or false. The memory length can be adjusted according to the type of RNN node used, the longer the memory, the longer the dependency that the RNN can learn. LSTM (Long Short-Term Memory) is an improved model of RNN that solves the original gradient explosion problem and is more commonly used in currently anomaly detection. CNN has better results in extracting spatial features [5], and its unique convolutional layer can continuously extract larger spatial features. Reference [6] introduces some traditional machine learning and deep learning methods applied in network security detection, such as SVM (Support Vector Machines), K-nearest neighbour, decision trees, DBN (Deep Boltzmann Machine) and so on, as well as commonly used publicly datasets. All these methods have good applications in anomaly detection, but individual models always have some shortcomings and therefore research in recent years has rarely used them individually.

In reference [7] an anomaly detection scheme based on hybrid deep learning is proposed for anomalous data stream detection in social multimedia environments to improve the reliability of SDN (Software Defined Network). The scheme uses an improved RBM and a gradient descent based SVM to detect anomalous activities, with a newly introduced Dropout in the RBM and an improved SVM by encapsulating hybrid kernel functions and gradient descent methods. In addition, the method uses end-to-end data transfer to meet the requirements of high bandwidth and low latency in realistic environments. The proposed scheme was eventually evaluated on three datasets, TITE, KDD 99 and CMU, and its detection rate exceeded 99% in all cases, proving its effectiveness and efficiency. The approach combines deep learning models with traditional machine learning models, which is a major direction for current applications.
In reference [8], a deep hierarchical network is proposed to detect malicious traffic in the network. The network uses 1D-CNN to extract spatial features and GRU to extract temporal features and is validated on the ISCX2012, USTC-TFC2016 and CICIDS2017 datasets respectively. Reference [9] also uses spatial and temporal features for anomaly detection, with the difference that it uses a 2D-CNN and a bi-directional LSTM model, and the detection object changes from a sequence-based data stream to anomalies in the video. The model extracts deep CNN features from consecutive frames and then passes the extracted deep features to a multi-layer bidirectional LSTM. The model can accurately classify anomalous/normal events that are occurring in the surveillance scene. The improvement was 3.41 percentage points on the UCF-Crime dataset and 8.09 percentage points on the UCFCrime2Local dataset. Spatial and temporal features are a major focus of current research and their effectiveness is very good, this paper is also a study from that perspective.

From the above it can be seen that the model has changed from a single model to a combined model, where the single model uses its own structure to extract and classify the samples, while the combined model uses a deep neural network as a feature extractor to obtain a rich feature representation [10], which is then fed into the next network, combining the advantages of both to improve the effectiveness of anomaly detection. To improve the effectiveness of the model different variant models have been generated which combine with each other to suit different needs, such as denoising auto-encoders, convolutional auto-encoders, etc. This paper is inspired by the variant structure of the model and uses a combination of the SAE variant model CAE and LSTM to enhance its benefits.

3. Model Based on a Combination of CAE and LSTM

3.1. Style and Model Overview

In this paper, a model combining multiple convolutional auto-encoders and a long short-term memory network is constructed. The convolutional auto-encoder and the long short-term memory network are in a back-and-forth tandem structure, and the use of multiple CAEs can extract different feature representations. The convolutional auto-encoder can accurately extract spatial features of samples, and the long short-term memory network can use the extracted features to obtain the temporal structure. Furthermore, the use of the convolutional auto-encoder to filter and downscale the sample features first can greatly improve the processing efficiency of the long short-term memory network.

The convolutional auto-encoders [11] is a modification of the stacked auto-encoder, so it has all the properties of the SAE, both of which can encode and decode the sample and try to make the output equal to the input. The Dense layer learns global patterns from the input features, while the convolutional layer learns local patterns from the features. In contrast to the Dense layer, the convolutional layer has two important properties: translation invariance, which means that after a feature is learned, it can be recognized at any subsequent position, which allows it to learn a generalized representation of the data with few samples; and spatial hierarchy of learned patterns, which is achieved by stacking the convolutional layers. CAE replaces the original Dense layer with a convolutional layer, which not only gives the model the advantage of SAE's comprehensive feature extraction with minimal loss function, but also has the advantages of translation invariance of the convolutional layer and learning the spatial structure of the feature. This makes the CAE better at processing effective features than the SAE alone.

The input sample is encoded with decreasing feature dimensionality, similarly to a compression operation, but the compressed features can be recovered by decoding, and the recovered output sample is approximately equal to the input sample, so the effective features contained in either layer of the convolutional auto-encoder are the same, only the degree of compression is different, and the removed features are mostly useless. In this paper, we use the sample features compressed by the encoder, which are not only comprehensive but also of lower dimensionality.

It should be noted that in actual simulation experiments the convolution layer faces the choice between 1D and 2D convolution, because the dataset used in this paper is a PCAP file, which belongs
to sequence type data, and 1D convolution has a better application for sequence type samples and 2D convolution has a better application for picture type data, so this paper uses a 1D convolution kernel.

The LSTM uses the valid features extracted from the encoder part of the convolutional auto-encoder to be input in sequential order of time. The neurons in each layer within the network memorize the information in the valid features for use in the current moment's output on the one hand, and on the other keep superimposing the information state to the next moment, until all features for that sample have been input. The final output is a vector that reflects the relationship between all the valid features of that sample. In contrast to the output of a CNN, which is only related to the current input data, the output of an LSTM is influenced by all the previous input data. This greatly strengthens the connection between features and allows for more accurate recognition and detection of samples.

3.2. Handling Process
The firstly is to train the CAE model, input the MNIST dataset into the CAE model and train it. The encoder part compresses the sample features by dimensionality reduction, and the decoder part decompresses the compressed features by dimensionality increase, using the characteristics of SAE to make the output equal to the input as much as possible, at which time the loss function value of the effective features is minimized and the parameters of the neurons in each layer have been trained. By freezing the encoder part, the compressed effective features can be obtained from the input samples.

Then, the encoder part is connected to the LSTM. The frozen encoder part of the CAE can obtain the compressed effective features, which contain many spatial features after the processing of the convolutional layer, and the features are arranged sequentially and contain temporal features, and these features are input to the LSTM to detect and classify the samples.

3.3. CAE Learning Space Features
Although the convolutional auto-encoder is used, it is also a convolutional neural network, only that it is compressed step by step. After the input size of (784,1) samples, the convolutional operation is performed first, and the output size is (784,16) feature map, i.e. the filter is 16, and then the dimensionality reduction is performed layer by layer according to the SAE building rule, where the convolutional kernel is set to 2, and its size after compression by the encoder The final size is (196,8), and the decoder uses upsampling to decompress the data, eventually recovering to (784,1). The entire model is then trained, intercepting the encoder part and freezing the learning space features for each parameter. By freezing, mean keeping the parameters in each layer unchanged and using that parameter to train the learning of new samples.

3.4. LSTM Learning Time Features
After the LSTM gets the data features from the encoder, the output is set to 196 neurons. Within each neuron the valid features are processed, and the neuron processes the input data at time t-1 to generate the output value ht-1 and the state value ct-1. These two values are saved until time t and are re-entered into that neuron along with the new input features. The output value ht and the state value ct are generated at time t and so on until all the data has been processed. In addition, the Dropout is set to 0.1 during processing to reduce overfitting, and the final output is sent to the Dense layer for detection and classification.

4. Simulation Experiments
4.1. Experimental Environment and Indicators
This dataset uses USTC-TFC2016, which contains 20 classes of data traffic. In order to evaluate the performance of the model, four parameters, namely accuracy, precision, recall and f1-score, were selected in this paper. Their formulas are as follows:

\[
\text{Accuracy} = \frac{(TP + TN)}{\text{total samples}}
\]
\[
\text{Precision} = \frac{TP}{(TP + FP)}
\]
Recall = TP / (TP + FN)
F1-score = 2*precision*recall/(precision + recall)

TP is a sample tested as positive and predicted as positive, FP is a sample tested as negative but predicted as positive, FN is a sample tested as positive but predicted as negative, TN is a sample tested as negative and predicted as negative, where the total sample is the sum of TP, TN, FP and FN.

In order to verify the superiority of CAE, this paper compares the CAE model with the SAE model to see the effect of the learning process by the loss value of both, the smaller the loss value the better the performance of the model. In the construction of the model, the same dataset is used and the number of layers and coded dimensions of the model are set to the same size.

4.2. Experimental Results and Analysis
From figure 1, we can see that the figures of CAE and SAE are very similar, their results are similar as one transfer to another, the biggest difference is that when Epochs is less than 20, the loss value of SAE has a period of very slowly decline, while CAE declines are rapidly, so the loss value of CAE is lower than the value of SAE. The difference between the two loss values is greatest when the Epochs are approximately equal to 20. The loss function used in this paper is binary cross-entropy, and the optimizer is Adadelta, so from the above results, it can be observed that CAE is better than SAE.

![Figure 1. CAE and SAE loss.](image)

Figure 2 shows the results obtained from the experiments on the dataset. From the graph, we can see that all four values have exceeded 99%, indicating that the detection effect is good and can identify most of the data. The f1-score shows the relationship between precision and recall, preventing one from being too large and one from being too small, and the performance of both precision and recall is better in this paper. In order to compare the performance of the models, this paper is also compared with other studies in table 1.

![Figure 2. Comparison of experimental results.](image)
Table 1. Comparison of study results.

| Model          | Accuracy | Precision | Recall  | F1-score |
|----------------|----------|-----------|---------|----------|
| [5]            | 99.17%   | 99.2%     | 99.23%  | 99.21%   |
| [7]            | 99.94%   | 99.09%    | 99.99%  | 99.53%   |
| this paper     | 99.98%   | 99.94%    | 99.3%   | 99.62%   |

All of the above studies used the USTC-TFC2016 dataset for the detection of abnormal traffic. As can be seen from the data above, this paper outperforms reference [5] on several parameters. Compared to reference [7], it is about 0.7 percentage points lower on Recall but 0.65 percentage points higher on Precision than the other side. The f1-score defines the relationship between Recall and Precision and it can be used to measure the strengths and weaknesses of the two together. The results show that this paper is slightly better than the other, and so is the accuracy. This indicates that this paper is better than the other study in terms of effectiveness.

5. Conclusions

In this paper, the anomaly detection method of network traffic is studied. In the current context, to improve the processing efficiency and detection effect, the use of convolutional auto-encoder network and long short-term memory network is proposed to jointly process the network data stream. The combination of the extracted features and the temporal features extracted by the long short-term memory network can be effective in detecting abnormal traffic. However, there are some shortcomings in the research process, mainly in two aspects: firstly, only one dataset is used in the training experiment, and the performance on other datasets is yet to be studied; secondly, the LSTM model takes more time to process the data and is not fast enough. The next step will be to try to combine it with traditional machine learning to ensure the accuracy and speed as much as possible.

References

[1] Kwon D, Kim H, Kim J, et al. 2019 A survey of deep learning-based network anomaly detection Cluster Computing 22 (1) 949-961.
[2] Chandola V, Banerjee A, Kumar V. 2009 Anomaly detection: A survey ACM computing Surveys (CSUR) 41 (3) 1-58.
[3] Berman D S, Buczak A L, Chavis J S, et al. 2019 A survey of deep learning methods for cybersecurity Information 10 (4) 122.
[4] Mahdavifar S and Ghorbani A A 2019 Application of deep learning to cybersecurity: A survey Neurocomputing 347 149-176.
[5] Wang W, Zhu M, Zeng X, et al. 2017 Malware traffic classification using convolutional neural network for representation learning 2017 International Conference on Information Networking (ICOIN) pp 712-717.
[6] Xin Y, Kong L, Liu Z, et al. 2018 Machine learning and deep learning methods for cybersecurity IEEE Access 6 35365-35381.
[7] Garg S, Kaur K, Kumar N, et al. 2019 Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in SDN: A social multimedia perspective IEEE Transactions on Multimedia 21 (3) 566-578.
[8] Wang B, Su Y, Zhang M, et al. 2020 A deep hierarchical network for packet-level malicious traffic detection IEEE Access 8 201728-201740.
[9] Ullah W, Ullah A, Haq I U, et al. 2021 CNN features with bi-directional LSTM for real-time anomaly detection in surveillance networks Multimedia Tools and Applications 80 (11) 16979-16995.
[10] Chalapathy R and Chawla S 2019 Deep learning for anomaly detection: A survey. arXiv preprint arXiv:1901.03407.
[11] Turchenko V, Chalmers E and Luczak A 2017 A deep convolutional auto-encoder with pooling-unpooling layers in caffe arXiv preprint arXiv:1701.04949.