Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/6/20230939

A deep learning model for accurate and robust internet traffic classification

Tianhao Fu[1,4,†], Boyuan Sun[2,†], Chenyue Zhang[3,†]

[1]Villanova College, King City, ON, L7B 0P5, Canada
[2]Chengdu Foreign Languages School, Chengdu, Sichuan, 611731, China
[3]Shanghai Nanyang Model Private School, Shanghai, 200032, China

[4]TFu@villanovacollege.ca
[†]These authors contributed equally

Abstract. Network traffic classification is significant due to the fast growth of the number of internet users. The traditional way of classifying the large number of traffic generated by these users is becoming less effective. Therefore, many researchers made a network traffic classifier based on deep learning. However, those classifiers do not provide far better results and perform poorly when dealing with encrypted information. This paper tries to approach highly accurate and robust results in both encrypted and unencrypted networks by using machine learning algorithms. The algorithm used is the convolutional neural network (CNN). The performance of the proposed CNN is compared with that of the classical LeNet-5 network. Experimental results show that the classifier based on the proposed CNN performed better when dealing with both encrypted and unencrypted datasets, achieving a maximum average accuracy of 83.55%. Moreover, it is not sensitive to hyper-parameter choices, indicating its superiority in robustness. Compared with traditional network classifiers, the network classifier based on CNN can improve accuracy and improve stability.

Keywords: Network traffic classification, deep learning, convolutional neural network.

1. Introduction

With the fast increase of global Internet users and the development of the Internet, the prevention of leakage and security protection of privacy and safety on the Internet has become a major concern of Internet users [1-3]. Communications over the Internet are easily intercepted by hackers, so online users are constantly at risk of being bugged or having their data stolen. Information leaks and cyber-attacks have led to a significant increase in public awareness of cyber security, as well as a growing awareness of data protection [4]. For internet users, data leakage can invade the privacy of important data; Meanwhile, data leakage can easily lead to the theft of important information belonging to the government, jeopardizing national network security [5]. In order to secure data on the Internet, encryption algorithms for traffic are rapidly developing. By 2020, more than 80% of traffic is encrypted by encryption algorithms, and the fact that encryption technology hides traffic information undoubtedly benefits the public [6]. However, encryption technology has also posed a significant challenge to network inspection, as it has made it impossible for technicians to detect traffic without decrypting it while keeping users safe. Accurate control of the network environment is not possible without accurate

DOI: 10.54254/2755-2721/6/20230939

detection. This shows that encryption technology is a double-edged sword. Encryption hides data information from eavesdropping or data theft, but at the same time, it also hides malicious or security-hazardous network traffic [7].

The fine-grained classification of network traffic is the basis for important tasks. As network traffic is increasing, network attacks are emerging and network security issues are becoming increasingly serious. It is crucial to implement the right management strategy for network resources and to identify different types of applications. In recent years, researchers in the field of network security have developed a number of different traffic classification methods to suit different application types and multiple network scenarios. The first is the conventional traffic classification method, which is two folds, the former is a port-based method and the latter is a load-based method. However, when dealing with vast amounts of network traffic data that is encrypted by cryptographic algorithms, these methods are unable to easily handle that encrypted network traffic. The new generation of traffic classifiers rely on machine learning methods that classify traffic based on statistical or time-series characteristics (e.g., flow duration, forward arrival interval), making it possible to classify both encrypted and unencrypted traffic. While such methods address the problems of encryption and reduced classification accuracy that traditional traffic classification methods cannot handle, they face new challenges in designing appropriate features. As a result, there is some recent work to identify encrypted traffic based on deep learning methods, and high performance has been achieved with the use of deep learning [8].

This paper will build on the traffic classification method described above and make improvements based on CNN networks [9], adjusting the number of CNN network layers to achieve the goal, of accurate and effective classification of encrypted and non-encrypted network traffic.

This study will focus on comparing the performance with that derived from CNN networks with different structures and with or without max-pooling. Based on the USTC-TFC2016 dataset, after various CNN networks are trained, this paper will explore the learning rate with the best performance based on the accuracy presented in the validation set and identify the CNN network with the best structure Ultimately, this paper will implement a network traffic classifier that performs well on encrypted traffic.

2. Method

2.1. Dataset

USTC-TFC2016 [10], the dataset used in this paper, stores a series of traffic datasets divided into two categories: Benign and Malware. The Benign dataset contains data streams of typical daily applications such as Gmail, Weibo, and Outlook. In addition, the Malware dataset includes botnets caused by Geodo's Phishing Attack, banking worms from CRIDEX, Miuref Trojan, Tinba Trojan, and Zeus Trojan. The traffic dataset is analyzed to determine whether the packets in USTC-TFC2016 are Benign or Malware.

2.2. CNN overview

This essay focuses on using the CNN approach to analyze traffic datasets, also known as Convolutional Neural Networks. Deep learning has become a popular topic in recent years. It is the construction of a neural network model, which is divided into input units, hidden units, and output units. After backpropagation was introduced in 1985, the neural network improved by calculating the cost function by linear regression, which means calculating the discrepancy between the actual and the predicted results, and then using bias derivatives to calculate the gradient and update the weights backward to optimize the model. The most effective and popular way to obtain a smaller cost function value is to use gradient descent, which is a way to find the minimum of the loss function by choosing the appropriate learning rate and making the loss function go down the hill. Note that if the hill is not concave, there may be more than one minimum due to different paths on the way down, so the result of gradient descent may be a local optimum.

CNN, as one of the main methods of deep learning, is most often used for image-related tasks, such as object detection and recognition. The main architecture of CNN is composed of different kinds of

DOI: 10.54254/2755-2721/6/20230939

layers, including convolution layers, downsampling layers, full connection layers, activation layers, and output layers. It will first automatically generate convolution kernels based on the existing data and labels in the dataset. Different convolution kernels will generate different features. Next, the data will be convolved, in which the convolution kernel and the numbers in the data set are multiplied and added together. The new numbers are filled into a new tensor, and the output is named a feature map. In addition, the Stride determines how often the Kernel moves, so choosing the proper Stride will avoid losing much information.

Then the data will go through the activation function, which compresses an actual number into a range to prevent gradient explosions, such as RELU and Sigmoid. After several convolutional layers and activation functions, the obtained data will enter the pooling layer (max pooling, avg pooling) to compress the image. This operation will be repeated several times. Finally, the Full connection layer will flatten the extracted features into a one-dimensional array, which integrates the extracted features, then compares the feature arrays with the existing data, and finally give the probability for different results. This is the CNN brief introduction. After training the model, overfitting or underfitting problems often occur. In the case of overfitting, the algorithm can be regularized to reduce the complexity of the model. In addition, the idea of dropout can be used to keep part of the data out of the computation.

2.3. Data Transformation

Since CNN is a type of image classifier, the traffic input must be converted to images first. Before the dataset is fed into our end-to-end neural network, each byte of the traffic is converted to a decimal integer from 0-255 and filled into a square grayscale map with a side length of 28. If the length is exceeded, the excess is discarded; if the length is insufficient, the process is repeated from the beginning at the current position instead.

2.4. CNN Model

2.4.1. LeNet-5. The original LeNet-5 [11], used in the experiments, has two two-dimensional convolutional layers. These layers are activated by a RELU layer and a max pooling layer. There are three linear full connection layers, followed by a RELU layer, which are leveraged to produce the probability distribution (13 classes).

Specifically, the first 3 layers include a convolutional layer with one input and six output channels, with a kernel size of 5, activated by the RELU layer and followed by a max pooling layer with size 2. The second 3 layers are the same. The output component consists of a fully connected layer (linear layer) and a RELU layer both repeat 3 times.

2.4.2. Proposed model. The proposed network consists of three two-dimensional convolutions. Each of them is followed by batch normalization and RELU, and finally, a linear full connection layer is passed to obtain the probability distribution (13 classes).

In this network, the first three layers contain a convolutional layer with 1 input and 16 output channels, with a kernel size of 5. They are subsequently connected with batch normalization and RELU. The second and third 3 layers are similar except for different parameters, which are the second convolutional layer has 16 input and 32 output channels, and a kernel size of 3. The third one has 32 input channels, 16 output channels, and a kernel size of 3. The output component is a fully connected layer (linear layer).

The proposed network structure has 11693 parameters (same as trainable parameters) in total, which is more lightweight compared to LeNet-5 as it contains 44681 parameters (same as trainable parameters).

3. Result

3.1. Experiment configuration

To validate the difference between the two networks, a total of 60 groups were included in the experiment, 30 of which used Proposed CNN and the other 30 groups used LeNet-5. For either of the

DOI: 10.54254/2755-2721/6/20230939

two 30 groups, the following training configuration is used. The learning rate is 0.01, 0.05, 0.1, 0.15, and 0.5 as a cycle, and each item is repeated once in the cycle. The batch size is 32, 64, and 128 as a cycle, and each item is repeated 5 times in the cycle. The number of batches is 2000 and 8000 as a cycle, and each item is repeated 15 times in the cycle. For all the 60 groups, the cross-entropy loss is leveraged as the loss function and SGD as the optimizer (weight decay = 0). 10,000 items from each class of USTC-TFC2016 were excluded for testing and the rest for training. In which the entire training set is shuffled in runtime, and then the corresponding number of batches according to the configuration of each group of the experiment are selected in turn.

As for the testing, for every group of the experiment, the model is tested by 10000 batches with a batch size of 1. The accuracy is calculated by $acc = \frac{c}{n}$ where c equals the number of correct classifications and n is the total number tested, which is 10000. It is one of the most seen methods for evaluating classification results. Moreover, the standard deviation (STD) is leveraged to measure the stability of the performances on different hyper-parameters.

3.2. Performance

Table 1 and Table 2 show the accuracy of the proposed network and LeNet-5 in different batch sizes respectively. Table 1 demonstrates the accuracy and the standard deviation with the training epochs of 2000. Table 2 displays that with the training epochs of 8000.

Table 1. Accuracy results with training epochs 2000.

Batch size	Learning Rate	Proposed Network ACC	LeNet-5 ACC
64	0.1	79.62%	57.62%
64	0.15	82.89%	66.73%
64	0.05	82.42%	53.77%
64	0.5	81.24%	7.70%
64	0.01	82.75%	9.44%
128	0.1	83.17%	60.52%
128	0.15	83.91%	65.67%
128	0.05	83.45%	33.63%
128	0.5	83.34%	8.01%
128	0.01	82.92%	19.09%
32	0.1	84.10%	54.07%
32	0.15	84.77%	47.79%
32	0.05	84.33%	50.92%
32	0.5	78.90%	54.17%
32	0.01	79.56%	7.88%
Average±STD		82.49%±1.77%	39.80%±22.19%

DOI: 10.54254/2755-2721/6/20230939

Table 2. Accuracy results with training epochs 8000.

Batch size	Learning Rate	Proposed Network ACC	LeNet-5 ACC
64	0.1	83.10%	80.03%
64	0.15	83.59%	8.04%
64	0.05	82.27%	58.52%
64	0.5	79.97%	66.75%
64	0.01	82.51%	7.82%
128	0.1	83.98%	61.59%
128	0.15	84.57%	7.80%
128	0.05	83.05%	76.12%
128	0.5	84.62%	7.26%
128	0.01	84.04%	36.14%
32	0.1	83.39%	68.31%
32	0.15	84.59%	75.16%
32	0.05	84.95%	49.10%
32	0.5	84.43%	57.89%
32	0.01	84.26%	36.69%
Average±STD		83.55%±1.24%	46.48%±26.37%

Based on the result, LeNet-5 performs poorly in most cases and is unable to classify effectively, but in certain cases, it is able to achieve extremely high accuracy. In contrast, although the maximum accuracy of the proposed network is not as good as that of LeNet-5, it performs consistently in most cases, extracts and differentiates features accurately, and is less sensitive to the configuration. Overall, the proposed network outperforms LeNet-5 in the task of classifying encrypted network traffic but still needs to be improved in terms of maximum accuracy.

4. Conclusion

In this paper, the network traffic classification problem is explored. To sum up, the performances of the two networks are compared on the USTC-TFC2016 dataset. The former is the proposed neural network without any max pooling operation, the latter is the classical LeNet-5. Measured by accuracy, the proposed one outperforms the LeNet-5 model. With a batch size of 2000, the average accuracy of the proposed network is 82.49%. It outperforms the LeNet-5. Moreover, with batch size 8000, the performance is 83.55%. Besides the accuracy, the proposed one is quite stable when using different hyper-parameters. The standard deviation of the proposed network is much smaller, indicating the performance is insensitive to hyper-parameters. This superiority could be partly attributed to the abundance of the max pooling operation. In traditional computer vision research, it is quite useful since the information that lies in natural images is sparse. However, in the Internet traffic classification problem, the information is dense and operations like max-pooling tend to omit valuable detailed information. In the future, more advanced CNN architectures could be explored to validate their effectiveness on Internet traffic classification problems.

References

[1] Fahad, A., Tari, Z., Khalil, I., Habib, I., & Alnuweiri, H. (2013). Toward an efficient and scalable feature selection approach for internet traffic classification. Computer Networks, 57(9), 2040-2057.

[2] Yuan, R., Li, Z., Guan, X., & Xu, L. (2010). An SVM-based machine learning method for accurate internet traffic classification. Information Systems Frontiers, 12(2), 149-156.

[3] Young, A. L., & Quan-Haase, A. (2013). Privacy protection strategies on Facebook: The Internet

DOI: 10.54254/2755-2721/6/20230939

privacy paradox revisited. Information, Communication & Society, 16(4), 479-500.
[4] Sun, C. C., Hahn, A., & Liu, C. C. (2018). Cyber security of a power grid: State-of-the-art. International Journal of Electrical Power & Energy Systems, 99, 45-56.
[5] Alneyadi, S., Sithirasenan, E., & Muthukkumarasamy, V. (2016). A survey on data leakage prevention systems. Journal of Network and Computer Applications, 62, 137-152.
[6] Rezaei, S., & Liu, X. (2019). Deep learning for encrypted traffic classification: An overview. IEEE communications magazine, 57(5), 76-81.
[7] Papadogiannaki, E., & Ioannidis, S. (2021). A survey on encrypted network traffic analysis applications, techniques, and countermeasures. ACM Computing Surveys, 54(6), 1-35.
[8] Pacheco, F., Exposito, E., Gineste, M., Baudoin, C., & Aguilar, J. (2018). Towards the deployment of machine learning solutions in network traffic classification: A systematic survey. IEEE Communications Surveys & Tutorials, 21(2), 1988-2014.
[9] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., et. al. (2018). Recent advances in convolutional neural networks. Pattern recognition, 77, 354-377.
[10] Wang, W., & Lu, D., (2016). USTC-TFC2016, URL: https://github.com/yungshenglu/USTC-TFC2016
[11] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.