Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH **Open Access**

Identification of encrypted and malicious network traffic based on one-dimensional convolutional neural network

Yan Zhou[1], Huiling Shi[1*], Yanling Zhao[2], Wei Ding[1], Jing Han[1], Hongyang Sun[1], Xianheng Zhang[1], Chang Tang[3] and Wei Zhang[1*]

Abstract

The rapid advancement of the Internet has brought a exponential growth in network traffic. At present, devices deployed at edge nodes process huge amount of data, extract key features of network traffic and then forward them to the cloud server/data center. However, since the efficiency of mobile terminal devices in identifying and classifying encrypted and malicious traffic lags behind, how to identify network traffic more efficiently and accurately remains a challenging problem. We design a convolutional neural network model: One-dimensional convolutional neural network with hexadecimal data (HexCNN-1D) that combines normalized processing and attention mechanisms. By adding the attention mechanism modules Global Attention Block (GAB) and Category Attention Block (CAB), network traffic is classified and identified. By extracting effective load information from hexadecimal network traffic, our model can identify most categories of network traffic including encrypted and malicious traffic data. The experimental results show that the average accuracy is 98.8%. Our model can greatly improve the accuracy of network traffic data recognition.

Keywords Network traffic identification, Convolutional neural network, Attention mechanism, Traffic Data format conversion

Introduction

Recent years have witnessed the development of Cloud Computing, the Internet of things (IoTs) and even the conceptual Internet of Everything (IoE) [1, 2], and intelligent application terminals in modern world are also developing in coherence with such trend. How to conduct real-time data analysis with limited networking and computing resources, and how to conduct network traffic analysis and classification via edge\cloud computing devices, which poses new challenges to network traffic monitoring and related issues. Firstly, cybersecurity must possess the capability to identify and block intrusive traffic data [3, 4]. Network security analysis of network traffic mainly involves identification of malicious network traffic to prevent malicious network attacks resulting in significant economic losses [5, 6]. Secondly, identification and classification of network traffic with higher accuracy

*Correspondence:
Huiling Shi
shihl@sdas.org
Wei Zhang
wzhang@sdas.org
[1] Shandong Provincial Key Laboratory of Computer Networks, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250000, Shandong, China
[2] Faculty of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250300, Shandong, China
[3] School of Computer Science, China University of Geosciences, Wuhan 430074, China

will improve network Quality of Service (QoS) and enable efficient traffic monitoring followed with effective resource allocation. Thirdly, network traffic identification can also be applied to Industrial Internet, Internet of Things (IoT), and cloud/edge network systems as well [7]. B. He et al. [8] proposed an intelligent VNFs configuration framework to solve the problem of network resource scheduling in CoT. Accurate traffic identification has been recognized as a crucial technology for improving Quality of Service (QoS) of the network [9, 10]. To solve these problems mentioned above, many approached and works focusing on network traffic identification have been proposed. Among these proposed approaches, those based on machine learning have the most promising prospects in general [11]. With the increasing number of terminal devices in the Internet of Things, the need for more efficient use of shared computing and communication resources in an end-to-end edge-cloud environment becomes more urgent [12].

According to the recently published works, the methods for identifying network traffic mostly focused on the training of machine learning models, such as convolutional neural networks, recurrent neural networks, decision trees, etc. Although machine learning models can greatly accelerate the extraction of network traffic characteristics [13], most existing solutions do not take into account the processing of the network traffic data format itself.

Our contributions are as follows:

1. In the model data pre-processing process, we introduced the normalization module to resolve the problem of insufficient and unbalanced data distribution caused by the small difference of network traffic categories.
2. In the model design process, we introduced the adjusted Global Attention Block (GAB) and Category Attention Block (CAB) to deal with more detailed data information of encrypted traffic and malicious traffic category.
3. We designed four different network experimental environments to identify conventional traffic, encrypted traffic, malicious traffic, mixed traffic, etc., so that it can detect and classify different network traffic categories more efficiently. The classification results are compared with other more advanced methods. The results show that our model can identify and classify network traffic data categories with higher precision.

The rest of this paper is structured as follows: The second section discusses the related work. The third section introduces the data preprocessing. The fourth and fifth sections respectively introduce the convolutional neural network model [14] HexCNN-1D and the batch normalization and attention mechanism module we added after adjustment, respectively. The sixth section presents the experimental data and index setting. The seventh section analyzes the experimental results. Section eighth discusses our future work and improvements.

Related work

There are four main network traffic classification methods [15–17]: methods based on port identification [18], methods based on deep packet detection [19], methods based on statistical processing [20], and methods based on user behavior [21]. Most network protocol ports are based on security policies; thus, the identification accuracy of such port identification-based methods is relatively low, and the deep packet detection-based method cannot process the current encrypted network traffic data. With the ubiquitous usage of machine learning [22–24], researchers are investigating approaches based on statistical processing and behavioral norms.

Traditional network traffic classification methods include clustering, support vector machines, C4.5 Decision Tree (C4.5), and etc. Most of these traditional methods have low accuracy and low classification efficiency. Anshu Priya et al. [25] proposed the use of K-Means clustering algorithm to analyze real-time network data traffic situations in universities. However, the clustering algorithm has poor classification efficiency for data categories with high similarity. Wang et al. [26] used C4.5 to classify P2P traffic, which is used to describe the behaviour characteristics of applications. The disadvantages of the C4.5 algorithm is that the training time is long and it is only suitable for processing small data sets. Coull et al. [27] proposed to classify p2p traffic by analyzing packet features and proposed traffic analysis of encrypted messaging services: Apple iMessage and other message classification. Mauro et al. [28] proposed to uncover encrypted WebRTC traffic by machine learning tools, using the random forest approach. Traditional feature-based statistical classifiers are becoming less suitable for today's massive data processing.

Deep learning has been gradually hybrided with more research fields to generate more efficient and appliable network models thanks to its powerful function extraction capability and efficient model parameter calculation. Segun I. Popoola et al. [4] proposed a deep neural network to classify network traffic in the scenario of Internet of Things, aiming at Zero-Day Botnet Attack Detection. However, the time cost of training model is large, so it cannot be applied to large-scale data. Shi

Dong et al. [29] proposed an optimization method for abnormal network traffic detection based on a semi-supervised double-depth Q-network (SSDDQN). Based on the above, Shi Dong [30] proposed an improved support vector machine (SVM) algorithm, the cost-sensitive support vector machine (CMSVM), to solve the imbalance problem in network traffic identification. Wang et al. [31, 32] for feature extraction from raw traffic data after preprocessing in two dimensions of CNN-1D and CNN-2D. The authors demonstrated the superiority of these two methods by observing and elaborating the accuracy scores achieved in the experimental evaluation metrics, etc. Lotfollahi et al. [33] proposed Deep Packet: a new method for cryptographic traffic classification using deep learning. However, the shortcoming of deep packet detection technology is obvious, it is vulnerable against the same kind of network attacks, and the deployment of deep packet detection is difficult, lest additional burden on the processor. Zou et al. [34] proposed a method for cryptographic traffic classification method based on convolutional Long Short-Term Memory (LSTM) neural networks. However, after a long period of training and increasing of the number of layers, the problem of gradient explosion is easily encountered. Bu et al. [35] proposed a deep parallel network (NIN) neural network model. Since its introduction, Deep learning has played an increasingly important role in machine learning. Convolutional neural networks (CNN), recurrent neural networks (RNN), and long and short-term memory network (LSTM) models gained their recognition for their excellent performance in the field of computer vision.

There are many common traffic classification methods, each with its own advantages and disadvantages. For example, port number-based classification is the easiest to implement, but has low identification accuracy and limited applicability. The classification method based on deep packets has a high accuracy but cannot detect encryption services. Therefore, future research will focus on network traffic classification and identification using machine learning methods. As a part of machine learning, researchers are trying to apply deep learning to the field of network traffic recognition technology. In this paper, a lightweight neural network model is proposed to identify classified network traffic data types.

Network traffic data pre-processing

Hexadecimal data of network traffic conversion

The ISCX-VPN-NonVPN-2016 and USTC-TFC2016 datasets are used in this paper. As shown in Table 1, we selected the following nine data streams by category in the ISCX-VPN-NonVPN-2016 dataset: AIM, Facebook, Email, Netflix, Hangouts, YouTube, Skype, Vimeo, and Spotify, and packets corresponding to the nine data steams encapsulated by the VPN.

Table 1 The data used In ISCX-VPN-NonVPN-2016

Class Option	The Numerical	Class Option	Numerical
AIM	4869	VPN-AIM	5000
Email	5000	VPN-Email	5000
Facebook	5000	VPN-Facebook	5000
Hangout	5522	VPN-Hangout	5016
Netflix	5000	VPN-Netflix	5031
Skype	5000	VPN-Skype	5009
Spotify	5000	VPN-Spotify	5022
Vimeo	5000	VPN-Vimeo	5014
YouTube	5000	VPN-YouTube	5000

Table 2 The data used In USTC-TFC2016

Class Option	The Numerical
BitTorrent	5000
Facetime	5000
Gmail	5272
MySQL	5000
World Of Warcraft	5000
Weibo	5001
Skype	5000
Virut	5035
Nsis-ay	5058
Zeus	5004

As shown in Table 2, we selected the 7 + 3 category in the USTC-TFC2016 dataset. Among them, there are seven different types of regular network traffic: BitTorrent, Facetime, Gmail, MySQL, Skype, Weibo, and World of Warcraft, and three different types of malicious network traffic: Zeus, Virut, and Nsis-ay. Above table shows the selected network traffic data types along with volume statistics.

We find that the effective content output in hexadecimal form in each PACP packet in the two datasets has obvious characteristic features, and most of the effective bytes in the packet are between [50, 1480] bytes.

Therefore, for the data flows captured in the dataset described above, we store approximately 5000 pieces of data in hexadecimal format for each type of data flow. Each data flow collects 1480 bytes of packet load through the preprocessing model. If the payload length is less than 1480 bytes of traffic, we use complement 0 to expand it to 1480 bytes for storage.

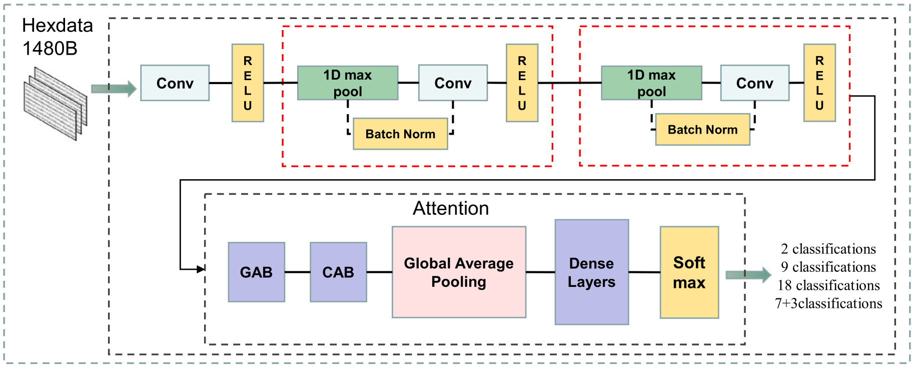


Fig. 1 HexCNN-1D Model Flow Chart

Network traffic identification framework

Convolutional neural network architecture

The design process of the deep learning network model are proposed in this section. The original flow data is first input into the preprocessing module, and then output data that can be directly used by the convolutional neural network via four steps: header information processing [36], key information extraction, and data reprocessing. The preprocessed training data is then fed into the deep learning network training module [37], where the convolutional neural network model is trained through feature extraction, data simplification, category judgment, and feedback adjustment successively. Finally, the test data is fed into the test module, which contains the trained convolutional neural network model, and the system is evaluated and elaborated based on the classification results.

HexCNN-1D model structure design

The one-dimensional convolutional neural network (HexCNN-1D) workflow is based on the network traffic recognition method. The input data of the model are the hexadecimal data obtained after preprocessing. After training the model, the network traffic identification work is completed according to the different traffic categories.

To prevent overfitting, we added an attention mechanism and a batch normalization layer to the design of the HexCNN-1D model. Normalization returns an uneven distribution to a normalized distribution. This allows the processing data to be distributed into sensitive regions of the activation function, speeding up model training and preventing gradient disappearance.

The flow of the HexCNN-1D algorithm based on a convolutional neural network is shown in Fig. 1.

Batch normalization and attention mechanism addition

Considering the large amount and load of network traffic data to be processed, the traditional one-dimensional convolutional neural network model design cannot meet the lightweight requirement of identifying the types and categories of encrypted and malicious traffic with higher accuracy. Therefore, we add a normalized processing module and an attention mechanism module within our model.

Batch normalization addition

When designing the convolutional neural network model, the Batch Normalized (BN) module is considered as an addition to the normal convolutional neural network model [38]. The BN module can solve problems such as slow convergence rates and gradient saturation caused by internal covariate shift [39].

$$y_i^{(b)} = BN(x_i)^{(b)} = \gamma\left(\frac{x_i^{(b)} - \mu(x_i)}{\sqrt{\sigma(x_i)^2 + \epsilon}}\right) + \beta \qquad (1)$$

${x_i}^{(b)}$ represents the value of the $i-th$ input node of this layer when the $b-th$ sample of the current batch is input, x_i for $[x_i^1, x_i^2, x_i^3, \ldots, x_i^m]$ a row vector, length of batch size m, μ and σ for the mean and standard deviation, ϵ division by zero to prevent the introduction of a minimum quantity (negligible), β and γ for the shift and scale parameters.

Attention mechanism addition

Due to the uneven distribution of data, the model will pay more attention to sufficient data, which will affect the final classification effect. As mentioned in this paper [40], CBAM is a lightweight general module, that can be applied to any CNN model and plays a non-negligible role in the application of GAB and CAB [41]. GAB and CAB can be used to learn the recognition features, so as to better resolve the problem of low accuracy caused by uneven data distribution.

$$M_{c\_a} = (ReLU(Conv2(GAP(M_{G\_IN})))) \otimes M_{G-IN}, M \in R^{H\times W\times C}, M_{G-IN} \in R^{H\times W\times C'}, C' = C/2 \tag{2}$$

The channel attention feature M_{c_a} is calculated in Formula 2, where H denotes the height, W represents the width, C represents the number of channels, and $ReLU$ represents the use of ReLU activation function, GAP represents the global average pooling, M_{G-IN} denotes the use of 1×1 convolution layer to reduce the number of channels.

$$M_{G-OUT} = M_{c\_a} \otimes (ReLU(C\_G(M_{c\_a}))) \tag{3}$$

The number of channels required for each category is calculated by $M', M' \in R^{H\times W\times ck}$, where c is the number of channels needed to identify each category, and k is the number of classes. Half of the features are retained by $M''(M'' = M')$, and the Dropout function is removed to make a prediction with all the features.

Formula 3 calculates the output of GAB, namely the spatial attention feature map M_{G-OUT}, $M_{G-OUT} = M_{G-IN}$. M_{G-OUT} is used to store the subtle and different information of each network traffic category in the detailed network traffic data, which is used as the input to the subsequent CAB.

$$S_i = \frac{1}{n}\sum_{j=1}^{n} GMP(m_{ij}^{\varepsilon}), i = \{1,2,3,\ldots,k\}, S = \{S_1, S_2, S_3, \ldots, S_k\} \tag{4}$$

As can be observed in Formula 4, S_i represents the degree of significant response to the feature mapping of each category, GMP represents the global maximum pooling, m_{ij}'' represents the JTH feature of class i in M'' and the score S of each category of network traffic is calculated by averaging the sum of M'' maximum pooling.

$$M_{i\_avg}' = \frac{1}{n}\sum_{j=1}^{n} m_{ij}', i = \{1, 2, 3, \ldots, k\} \tag{5}$$

In Formula 5, M_{i_avg}' represents the feature output mapping feature map of the class i, and m_{ij}' represents the reaction of the JTH feature of the class i in M'. The sum of the characteristic fractions of each class is calculated and averaged.

$$A_{CAB} = \frac{1}{k}\sum_{i=1}^{k} S_i M_{i\_avg}', A_{CAB} \in R^{H\times W\times 1} \tag{6}$$

In Formula 6, A_{CAB} is to multiply and average the calculated scores of each class and the semantic features of the class. It helps to differentiate areas of DR Grading.

$$M_{C-OUT} = M_{C-IN} \otimes A_{CAB} \tag{7}$$

Finally, as shown in Formula 7, M_{C-OUT} is obtained by multiplying CAB and category attention A_{CAB}, enabling the model to obtain more accurate classification of different network traffic categories.

Experimental data and index setting

In this section, the public network datasets ISCX and USTC are used for experiments. The testing ratio of the training set was set to 7:3, and the sample set used in each experiment was described in detail.

Experimental metrics settings

In this research, four classification indices were used in the experiment: Accuracy, Precision, Recall, and F1-score. TP denotes the positive sample correctly predicted by the model, FN denotes the positive sample incorrectly predicted by the model, FN denotes the negative sample incorrectly predicted by the model, and TN denotes the negative sample correctly predicted by the model.

We use the ablation experiment and the confusion matrix [42] to validate the detection of different data traffic categories and the experimental results. Ablation experiments are commonly used in neural networks to learn about the network by deleting part of the network and studying its performance. The confusion matrix's function is to group the expected and actual results of all categories into the same table based on category. In this table, we can clearly observe the number of accurate and inaccurate recognitions for each category.

Dataset category classification

The ISCX dataset contains traffic characteristics and raw traffic (in PCAP format). In our experiment, the experimental environment was divided into two categories (VPN and non-VPN), nine and eighteen.

The UTSC dataset uses the class 7+3 (seven non-malicious traffic and three malicious traffic) categories in the UTSC dataset to determine the model's ability to detect malicious traffic. We have 1,000 of each, for a total of 10,000 samples. The experiment went through 50–60 iterations.

Table 3 Optimal hyperparameter setting

Hyper-Paramete	Value
Batch_size	20
Learning_rate	0.0001
Loss	Softmax_loss
Optimizer	Adam
Epochs	50

Configuration and parameter Settings

For hardware and software configuration, we have used python3, PC version of Windows 11, Processor 12th Gen Intel(R) Core (TM) i5-12500H 2.50 GHz, running memory 16.0 GB.

We iteratively optimized the hyperparameters of the model and conducted a lot of model tuning mainly for batch processing [43], optimizer, loss function, normalization operation, etc., as shown in Table 3 below, the optimal parameter settings of the model are provided. The Adam optimizer is capable of updating the model parameters by calculating gradient optimization. Softmax loss function, etc.

Experimental results of network traffic identification

Compared with HexCNN-1D methods

The following are the experimental findings of the HexCNN-1D convolutional neural network model in two classifications, nine classifications, eighteen classifications, and malicious and non-malicious classifications:

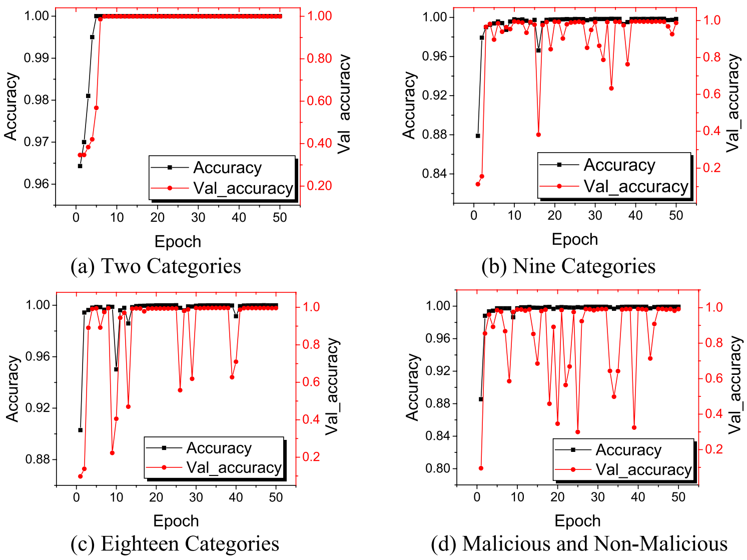
The HexCNN-1D model developed in this paper uses two different exposed data sets, as shown in Fig. 2, and the accuracy indices of all tests were kept above 98%.

As shown in Fig. 3, the above experimental results and data show that the HexCNN-1D model designed in this paper has a higher classification recognition accuracy and a more efficient classification effect.

Therefore, we suggest that the combination of a convolutional neural network and network traffic recognition can significantly improve the accuracy of network traffic classification technology and can be more successfully applied to network traffic detection.

As shown in Table 4, the USTC-TFC data set shows that the HexCNN-1D model has more than 98% identification accuracy against malicious traffic such as Zeus, Virut, and Nsis-ay. This shows that the HexCNN-1D model established in this paper possesses the capability to detect malicious traffic. The packet length of malicious traffic is longer than that of regular traffic. The model we designed can extract valid data fields and accurately identify different types of malicious traffic with limited packet length.

The deep learning convolutional neural network classification model HexCNN-1D was trained to extract different label features. Four independent scenario tests were set up to collect experimental data of the HexCNN-1D model and compare it with the classical machine learning model. As can be observed in Table 5, the model proposed in this paper is superior to other network machine learning models in identifying VPN and non-VPN traffic. Compared to the traditional model (Deep Packet, C4.5), the accuracy of our model



(a) Two Categories

(b) Nine Categories

(c) Eighteen Categories

(d) Malicious and Non-Malicious

Fig. 2 HexCNN-1D Mode Experimental Results

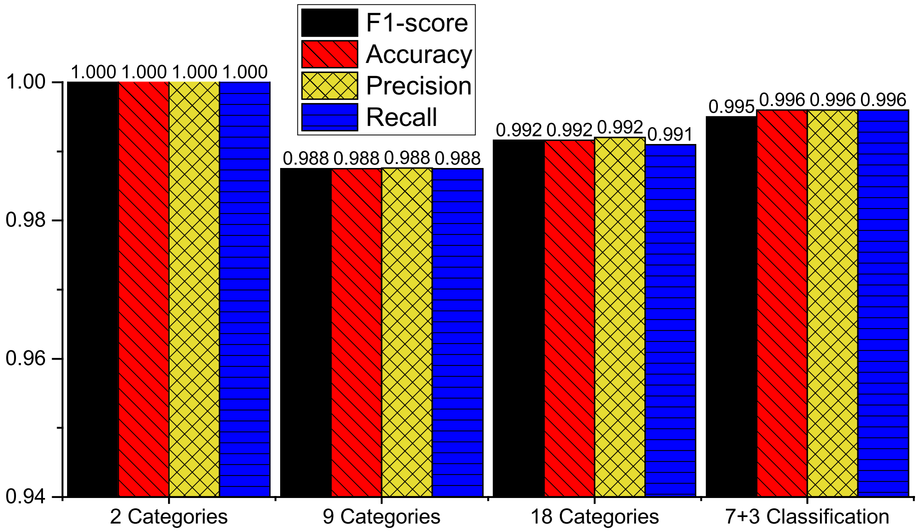


Fig. 3 Experimental Results of HexCNN-1D Model

is improved by 14% to 28%. Compared to the common 1D-CNN model, the accuracy of encapsulating network traffic in both Non-VPN and VPN is increased by about 3 percentage points.

Ablation experiments

In order to evaluate the effectiveness of the model by adding normalized processing and attention mechanisms, we performed ablation experiments on HexCNN-1D. As shown in Table 6, the model is mainly processed by a one-dimensional convolutional neural network, followed by modules for normalization processing and attention mechanism.

First, a single one-dimensional convolutional neural network was tested to calculate the Accuracy, Precision, Recall and F1-score of the model. Then, the accuracy of F1-score and other indicators of the model were increased by about 3% after the addition of normalized processing. Finally, CAB and GAB were added to the base model, and the overall index increased by about 2%, indicating that the attention module improved the efficiency of the model in identifying network traffic categories.

Table 4 Malicious traffic identification by HexCNN-1D

	Precision	Recall	F1-score
Zeus	99.8%	99.1%	99.3%
Virut	99.1%	98.4%	98.6%
Nsis-ay	98.1%	98.3%	97.9%

Table 5 Comparison with experimental results of different models

	Non-VPN		VPN	
	Precision	Recall	Precision	Recall
Deep Packet [31]	70.6%	70.6%	-	85.5%
C4.5 [17]	84%	87.6%	89%	85.5%
1D-CNN [32]	95.6%	95.6%	95.6%	95.6%
NIN(large) [24]	97.5%	97.4%	97.9%	97.9%
CNN-2D	98.7%	98.6%	98.6%	97.7%
HexCNN-1D	**98.8%**	**98.7%**	**98.8%**	**98.7%**

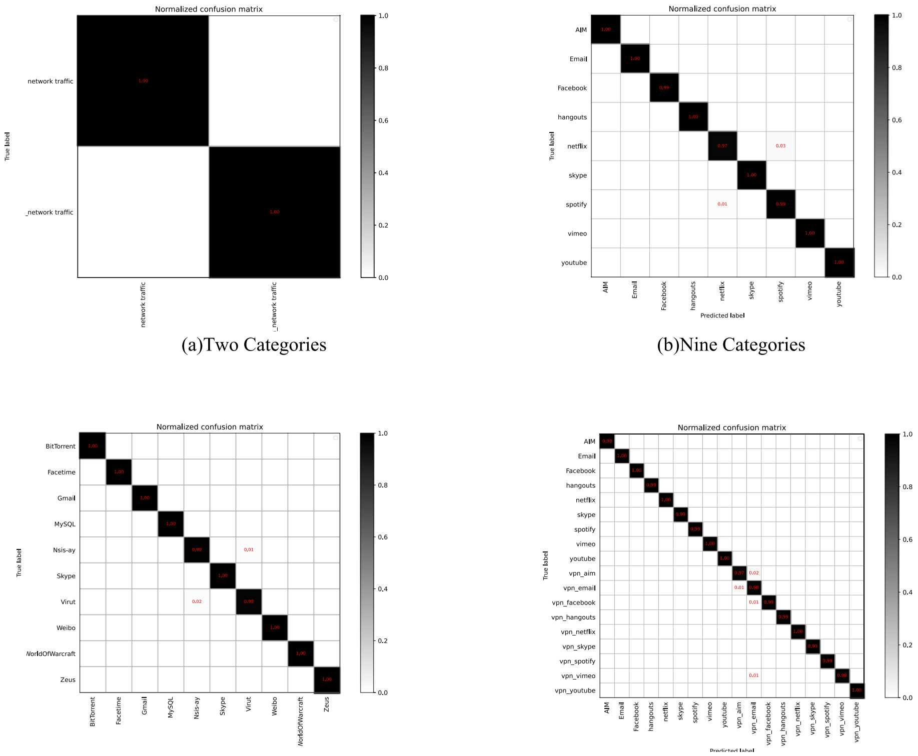
Confusion matrix validation experiment results

We used the confusion matrix shown in Fig. 4 to verify the experimental data and the classification accuracy of the experimental results.

The experimental results show that the HexCNN-1D classification model adopted in this paper has higher accuracy in four experimental scenarios, and has achieved excellent recognition results in the scenarios of encrypted traffic and malicious traffic identification.

Table 6 Comparison of ablation experiments

Model	Accuracy	Precision	Recall	F1-score
CNN-1D	90.1%	91.2%	92.7%	92.3%
CNN-1D + BN	95.2%	94.3%	94.6%	94.5%
CNN-1D + CAB + GAB	96.6%	96.7%	96.4%	96.6%
Our Model	**98.9%**	**98.8%**	**98.7%**	**98.7%**



(a)Two Categories

(b)Nine Categories

(c)Eighteen Categories

(d)Malicious and Non-Malicious

Fig. 4 The Obfuscation Matrix Verifies the Results

Conclusion

In this paper, a convolutional neural network model is designed to study network traffic recognition. In the data preprocessing stage, the influence of redundant information is ignored. The data preprocessing method was coupled with the convolutional neural network model designed by HexCNN-1D. Our model identifies traditional traffic data and VPN encapsulated traffic with an accuracy of 99%. We found that in the detection of malicious network traffic, such as Zeus, Virut and Nsis-ay, the accuracy of network traffic identification reached more than 98%. In the future, we will investigate the robustness of these models and the performance migration of the models under different flow modes.

Acknowledgements

The author would like to thank the anonymous reviewers for their valuable comments and the funding from Shandong Provincial Natural Science Foundation, Jinan Research Leader Workshop Project and Qilu University of Technology (Shandong Academy of Sciences) International Cooperation Pilot Project of Science, Education and Industry Integration Innovation.

Authors' contributions

All authors have participated in conception and design, or analysis and interpretation of this paper. All authors read and approved the final manuscript.

Funding

This work was supported by the Natural Science Foundation of Shandong Province (No. ZR2019LZH013) and the Natural Science Foundation of Shandong Province (No. ZR2020LZH010). Thanks to the Jinan Scientific Research Leader Studio Project under Grant (No. 2021GXRC091). Thanks to The Pilot International Cooperation Project for Integrated Innovation of Science, Education and Industry of Qilu University of Technology (Shandong Academy of Sciences) under Grant No.2022GH007.

Availability of data and materials

The corresponding author may provide the supporting data on request.

Declarations

Competing interests

The authors declare no competing interests.

Ethics approval and consent to participate

No ethical approval is required, and the authors express their consent to participate in the paper.

Consent for publication

Authors provide consent for publication.

Competing of interests

The authors have no relevant financial or non-financial interests to disclose.

Data Availability

The datasetsanalysed during the current study are available in the UNB and github repository,respectively: VPN-NonVPN Dataset (ISCXVPN2016) is: http://www.unb.ca/cic/datasets/vpn.html. The USTC– TFC 2016 dataset is:

https://github.com/echowei/DeepTraffic. All other data are available from the authorsupon reasonable request.

Received: 19 January 2023 Accepted: 22 March 2023
Published online: 10 April 2023

References

1. Zhou X, Xu X, Liang W, Zeng Z, Yan Z (2021) Deep-Learning-Enhanced Multitarget Detection for End–Edge–Cloud Surveillance in Smart IoT. IEEE Internet Things J 8(16):12588–12596. https://doi.org/10.1109/jiot.2021.3077449
2. Zhou X, Yang X, Ma J, Wang KIK (2022) Energy-Efficient Smart Routing Based on Link Correlation Mining for Wireless Edge Computing in IoT. IEEE Internet Things J 9(16):14988–14997. https://doi.org/10.1109/jiot.2021.3077937
3. H. Ahmed, A. Alsadoon, P. W. C. Prasad, N. Costadopoulos, L. S. Hoe and A. Elchoemi, "Next generation cyber security solution for an eHealth organization," 2017 5th International Conference on Information and Communication Technology (ICoIC7), pp. 1–5, 2017, doi: https://doi.org/10.1109/ICoICT.2017.807 4723.
4. Popoola SI, Ande R, Adebisi B, Gui G, Hammoudeh M, Jogunola O (2022) Federated Deep Learning for Zero-Day Botnet Attack Detection in IoT-Edge Devices. IEEE Internet Things J 9(5):3930–3944. https://doi.org/10.1109/jiot.2021.3100755
5. Ning J et al (2022) Malware Traffic Classification Using Domain Adaptation and Ladder Network for Secure Industrial Internet of Things. IEEE Internet Things J 9(18):17058–17069. https://doi.org/10.1109/jiot.2021.3131981
6. Kumar M, Mukherjee P, Verma K, Verma S, Rawat DB (2022) Improved Deep Convolutional Neural Network Based Malicious Node Detection and Energy-Efficient Data Transmission in Wireless Sensor Networks. IEEE Transactions on Network Science and Engineering 9(5):3272–3281. https://doi.org/10.1109/tnse.2021.3098011
7. Sun Q, Shi Y (2022) Model Predictive Control as a Secure Service for Cyber-Physical Systems: A Cloud-Edge Framework. IEEE Internet Things J 9(22):22194–22203. https://doi.org/10.1109/jiot.2021.3091981
8. He B, Wang J, Qi Q, Sun H, Liao J (2022) Towards Intelligent Provisioning of Virtualized Network Functions in Cloud of Things: A Deep Reinforcement Learning Based Approach. IEEE Transactions on Cloud Computing 10(2):1262–1274. https://doi.org/10.1109/tcc.2020.2985651
9. K. Yu, L. -z. Tan, X. -j. Wu and Z. -y. Gai, "Machine Learning Driven Network Routing," 2019 6th International Conference on Systems and Informatics (ICSAI), pp. 705–712, 2019, doi: https://doi.org/10.1109/ICSAI 48974.2019.9010507.
10. B. Yang and D. Liu, "Research on Network Traffic Identification based on Machine Learning and Deep Packet Inspection," 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), pp. 1887–1891, 2019, doi: https://doi.org/10.1109/ITNEC.2019.8729153.
11. S. Dong, P. Wang, and K. Abbas, "A survey on deep learning and its applications, " Computer Science Review, vol. 40, p. 100379, 2021, https://doi.org/10.1016/j.cosrev.2021.100379.
12. Zhou X et al (2023) Edge-Enabled Two-Stage Scheduling Based on Deep Reinforcement Learning for Internet of Everything. IEEE Internet Things J 10(4):3295–3304. https://doi.org/10.1109/jiot.2022.3179231
13. M. Li, D. Han, X. Yin, H. Liu, D. Li, "Design and Implementation of an Anomaly Network Traffic Detection Model Integrating Temporal and Spatial Features", Security and Communication Networks, vol. 2021, Article ID 7045823, 15 pages, 2021. https://doi.org/10.1155/2021/7045823.
14. A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar and L. Fei-Fei, "Large-Scale Video Classification with Convolutional Neural Networks," 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1725–1732, 2014, doi: https://doi.org/10.1109/CVPR.2014.223.
15. J. Zhao, X. Jing, Z. Yan, W. Pedrycz, "Network traffic classification for data fusion: A survey, " Information Fusion. pp. 22–47, 2021, https://doi.org/10.1016/j.inffus.2021.02.009.
16. Zhang J, Xiang Y, Wang Y, Zhou W, Xiang Y, Guan Y (2013) Network Traffic Classification Using Correlation Information. IEEE Trans Parallel Distrib Syst 24(1):104–117. https://doi.org/10.1109/tpds.2012.98
17. Velan, Petr, et al. "A survey of methods for encrypted traffic classification and analysis," International Journal of Network Management. pp. 355–374. 2015, https://doi.org/10.1002/nem.1901.
18. Y. Hu, D. -M. Chiu and J. C. S. Lui, "Application Identification Based on Network Behavioral Profiles," 2008 16th Interntional Workshop on Quality of Service, Enschede, Netherlands, pp. 219–228, 2008, doi: https://doi.org/10.1109/IWQOS.2008.31.
19. LiJuan Zhang, DongMing Li, Jing Shi and JunNan Wang, "P2P-based weighted behavioral characteristics of deep packet inspection algorithm," 2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering, Changchun, pp. 468–470, 2010, doi: https://doi.org/10.1109/CMCE.2010.5610457.
20. F. Risso, M. Baldi, O. Morandi, A. Baldini and P. Monclus, "Lightweight, Payload-Based Traffic Classification: An Experimental Evaluation," IEEE International Conference on Communications, pp. 5869–5875, 2008, doi: https://doi.org/10.1109/ICC.2008.1097.
21. Cao Z, Xiong G, Zhao Y, et al, "A survey on encrypted traffic classification," [C]//International Conference on Applications and Techniques in Information Security. vol. 490, pp. 73–81, 2014, https://doi.org/10.1007/978-3-662-45670-5_8.
22. Wu Q, He K, Chen X, Yu S, Zhang J (2022) Deep Transfer Learning Across Cities for Mobile Traffic Prediction. IEEE/ACM Trans Networking 30(3):1255–1267. https://doi.org/10.1109/tnet.2021.3136707
23. M. Mainuddin, Z. Duan, Y. Dong, S. Salman and T. Taami, "IoT Device Identification Based on Network Traffic Characteristics," GLOBECOM 2022 - 2022 IEEE Global Communications Conference, pp. 6067–6072, 2022, doi: https://doi.org/10.1109/GLOBECOM48099.2022.10001639.
24. T. Zheng and B. Li, "Poisoning Attacks on Deep Learning based Wireless Traffic Prediction," IEEE INFOCOM 2022 - IEEE Conference on Computer Communications, pp. 660–669, 2022, doi: https://doi.org/10.1109/INFOCOM48880.2022.9796791.
25. A. Priya, S. Nandi and R. S. Goswami, "An Analysis of real-time network traffic for identification of browser and application of user using clustering algorithm," 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), pp. 441–445, 2018, doi: https://doi.org/10.1109/ICACCCN.2018.8748706.
26. D. Wang, L. Zhang, Zhenlong Yuan, Y. Xue and Y. Dong, "Characterizing Application Behaviors for classifying P2P traffic," 2014 International Conference on Computing, Networking and Communications (ICNC), pp. 21–25, 2014, doi: https://doi.org/10.1109/ICCNC.2014.6785298.
27. Coull S E and Dyer K P, "Traffic analysis of encrypted messaging services: Apple imessage and beyond, " ACM SIGCOMM Computer Communication Review, pp. 5–11, 2014, https://doi.org/10.1145/267704 6.2677048.
28. M. Di Mauro and M. Longo, "Revealing encrypted WebRTC traffic via machine learning tools," 2015 12th International Joint Conference on e-Business and Telecommunications (ICETE), pp. 259–266. 2015, https://doi.org/10.5220/0005542202590266.
29. Dong S, Xia Y, Peng T (2021) Network abnormal traffic detection model based on semisupervised deep reinforcement learning. IEEE Trans Netw Serv Manage 18(4):4197–4212. https://doi.org/10.1109/TNSM.2021.3120804
30. S. Dong, "Multi class svm algorithm with active learning for network traffic classification, " Expert Systems with Applications, vol. 176, pp. 114885, 2021, https://doi.org/10.1016/j.eswa.2021.114885.
31. W. Wang, M. Zhu, J. Wang, X. Zeng and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 43–48, 2017, doi: https://doi.org/10.1109/ISI.2017.8004872.
32. Wang W, Zhu M, Zeng X, et al., "Malware traffic classification using convolutional neural network for representation learning," 2017 International conference on information networking (ICOIN). pp. 712–717, 2017, doi: https://doi.org/10.1109/ICOIN.2017.7899588.
33. Lotfollahi M, Jafari Siavoshani M, Shirali Hossein Zade R, et al., "Deep packet: A novel approach for encrypted traffic classification using deep learning," Soft Computing, pp. 1999–2012, 2020, https://doi.org/10.1007/s00500-019-04030-2.

34. Z. Zou, J. Ge, H. Zheng, Y. Wu, C. Han, and Z. Yao, "Encrypted Traffic Classification with a Convolutional Long Short-Term Memory Neural Network," 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), pp. 329–334, 2018, doi: https://doi.org/10.1109/HPCC/SmartCity/DSS.2018.00074.
35. Bu Z, Zhou B, Cheng P, Zhang K, Ling Z-H (2020) Encrypted Network Traffic Classification Using Deep and Parallel Network-in-Network Models. IEEE Access 8:132950–132959. https://doi.org/10.1109/access.2020.3010637
36. H. Zhou, Y. Wang, X. Lei, and Y. Liu, "A Method of Improved CNN Traffic Classification," presented at the 2017 13th International Conference on Computational Intelligence and Security (CIS), pp. 177–181, 2017, doi: https://doi.org/10.1109/CIS.2017.00046.
37. Samanta RK, Sanyal G, Bhattacharjee P, (2009) " Study and Analysis of Cellular Wireless Networks with Multiclass Traffic, " 2009 IEEE International Advance Computing Conference, pp. 1081–1086. https://doi.org/10.1109/IADCC.2009.4809164.
38. Kalayeh MM, Shah M (2020) Training Faster by Separating Modes of Variation in Batch-Normalized Models. IEEE Trans Pattern Anal Mach Intell 42(6):1483–1500. https://doi.org/10.1109/TPAMI.2019.2895781
39. M. Awais, M. T. Bin Iqbal, and S. H. Bae, "Revisiting Internal Covariate Shift for Batch Normalization," IEEE Trans Neural Netw Learn Syst, vol. 32, no. 11, pp. 5082–5092, Nov 2021, doi: https://doi.org/10.1109/TNNLS.2020.3026784.
40. S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module, " Computer Vision – ECCV, pp. 3–19, 2018, https://doi.org/10.1007/978-3-030-01234-2_1.
41. He A, Li T, Li N, Wang K, Fu H (2021) CABNet: Category Attention Block for Imbalanced Diabetic Retinopathy Grading. IEEE Trans Med Imaging 40(1):143–153. https://doi.org/10.1109/TMI.2020.3023463
42. J. L. Garcia-Balboa, M. V. Alba-Fernandez, F. J. Ariza-López and J. Rodriguez-Avi, "Homogeneity Test for Confusion Matrices: A Method and an Example," IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, pp. 1203–1205, 2018, doi: https://doi.org/10.1109/IGARSS.2018.851 7924.
43. I. Parashchuk and I. Kotenko, "Identification of the Traffic Model Parameters for Network and Cloud Platform Security Management," 2020 International Scientific and Technical Conference Modern Computer Network Technologies (MoNeTeC), Moscow, Russia, pp. 1–6, 2020, doi: https://doi.org/10.1109/MoNeTeC49726.2020.9258159.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.