Article

Not peer-reviewed version

Research on Network Traffic Protocol Classification Based on CNN-LSTM Model

Jiawei Jin * , Shuzhan Wang , Ziwei Liu

Posted Date: 23 April 2025

doi: 10.20944/preprints202504.1948.v1

Keywords: Network traffic analysis; Protocol classification; CNN-LSTM model; Feature selection; Machine learning

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 April 2025 doi:10.20944/preprints202504.1948.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Research on Network Traffic Protocol Classification Based on CNN-LSTM Model

Jiawei Jin [1,*], Shuzhan Wang [2] and Ziwei Liu [3]

[1] Technical University of Munich, Munich, German
[2] Amazon Web Services, Inc, USA
[3] University of Illinois, Urbana Champaign, Seattle, Washington, USA
* Correspondence: jiawei.jin@tum.de

Abstract: Network traffic analysis is essential for network security and performance optimization, yet classifying network traffic protocols remains a challenge. This study improves the classification and prediction of unknown network traffic protocols. By collecting and analyzing extensive traffic data we examine the correlation between traffic characteristics and protocol types.We introduce a CNN-LSTM model that integrates Convolutional Neural Networks (CNN) for local perception and weight sharing, and Long Short-Term Memory (LSTM) networks for temporal sequence modeling, which improves the accuracy of protocol classification. Experiments show that the CNN-LSTM model outperforms other models in terms of accuracy and F1 score. With feature selection, the accuracy reaches 0.981; while with raw features, both accuracy and F1 score reach 0.956. In contrast, standalone LSTM and CNN models show weaker performance and are more sensitive to changes in the number of features.This study validates the effectiveness of the CNN-LSTM model for network traffic protocol classification, and provides insights for future research. Future studies may explore ways to optimize the model structure and feature processing to cope with more complex network environment and traffic data.

Keywords: network traffic analysis; protocol classification; CNN-LSTM model; Feature selection; machine learning

1. Introduction

In today's digital age, network traffic analysis is very important for ensuring network security and optimizing network performance. With the increasing diversification and complexity of network applications, it is a challenging but indispensable task to accurately identify the protocol types in network traffic. Different network protocols correspond to different network behaviors and application scenarios. A deep understanding of the protocol composition and characteristics of network traffic can provide key support for network management, security monitoring and service quality optimization.

This study focuses on the field of network traffic analysis, aiming at accurately classifying and predicting unknown network traffic protocols by using machine learning technology. By collecting a large number of traffic data in a specific network environment, and deeply analyzing and modeling it, it is expected to dig out the internal relationship between network traffic characteristics and protocol types. The research results not only help to improve the accuracy and efficiency of network traffic analysis, but also provide valuable reference for further research and practice in related fields.

2. Literature Review

Network traffic classification plays a critical role in network management and security monitoring. Accurate identification of traffic protocols is essential for detecting anomalies, optimizing performance, and improving security measures. Over the years, numerous approaches,

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

ranging from traditional statistical methods to modern machine learning (ML) techniques, have been proposed to tackle this challenge. In this section, we review the state-of-the-art techniques and datasets for network traffic classification, with a focus on machine learning algorithms.

Azab et al. provide a comprehensive review of network traffic classification techniques, discussing both traditional and modern approaches, and identifying key challenges, including the dynamic nature of network traffic and the wide variety of protocols in use [1]. They emphasize that while older techniques, such as rule-based methods, have been widely used, machine learning algorithms are gaining prominence, due to their greater adaptability to evolving patterns and protocols. Shafiq et al. focus on a comparative analysis of ML algorithms for network traffic classification [2]. They demonstrate that supervised learning algorithms, such as decision trees, support vector machines (SVMs), and k-nearest neighbors (k-NN), outperform traditional methods in classification accuracy. Their work also highlights the importance of feature selection and tuning on algorithm performance. Similarly, Patel et al. conduct an analysis using multiple ML algorithms and show that Random Forest and SVM perform better in high-dimensional feature spaces compared to simpler classifiers [3].

Salman et al. further expand on these findings by reviewing ML-based approaches for Internet traffic classification [4]. Their study suggests that neural networks, particularly deep learning models, are well-suited for handling complex, high-volume traffic data, as they can automatically learn and extract valuable features from raw traffic data. They conclude that deep learning techniques, such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, are particularly effective for classifying network traffic with temporal and spatial dependencies.

Piskac and Novotny discuss the challenge of time-series data in network traffic, proposing time characteristics analysis as a way to classify traffic more effectively [5]. They argue that analyzing packet arrival times and inter-arrival times can help with distinguishing between different types of network traffic. This insight has been incorporated into more recent ML approaches, as shown by Soysal and Schmidt, who explore flow-based classification techniques using machine learning algorithms that leverage packet-level features and time-based statistics to improve classification accuracy [6].

The availability of diverse datasets is crucial for developing and evaluating traffic classification models. Azab et al. provide an overview of publicly available network traffic datasets used for training and testing network traffic classification models [7]. These datasets, which include traffic data from various applications, such as HTTP, FTP, and VoIP, are essential for evaluating the effectiveness of different algorithms. Dias et al. introduce a real-time network traffic classification approach, demonstrating that real-time processing of network data is feasible using the right feature extraction techniques and algorithms [8]. Their approach, combined with deep learning, enables fast and accurate classification in live network environments, providing a potential solution to the challenges posed by evolving network traffic.

Hwang R H et al. proposed a deep learning method based on LSTM networks to classify malicious traffic at the packet level [9]. Camelo M et al. proposed a general method for wireless network traffic classification based on deep learning [10]. Lopez-Martin M et al. constructed an Internet of Things network traffic classifier that combines CNN and cyclic neural network [11]. This classifier uses CNN to extract the spatial features of traffic, and circular neural network to mine the temporal features, thus realizing the efficient classification of Internet of Things network traffic. Nithya B et al. classified the main congestion control algorithms of Transmission Control Protocol (TCP) by using a convolutional neural network-long-short-term memory network (CNN-LSTM) model [12]. By leveraging the local feature extraction ability of CNN and the time series processing advantage of LSTM, their research accurately identifies the network traffic characteristics under different TCP congestion control algorithms.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 April 2025 doi:10.20944/preprints202504.1948.v1

3. Data Introduction

The data of this study is of great significance in the field of network traffic analysis, which provides a rich information base for model training and analysis. Data is collected in a specific network environment and obtained by capturing and processing network traffic, covering many aspects of information.

3.1. Data source

The data was collected in the network part of Universidad Del Cauca in Popayá n, Colombia. In this network environment, the network traffic in different time periods was comprehensively monitored and collected. Traffic statistics (IP address, port, arrival interval, etc.) are obtained by CICFlowmeter, and application layer protocols are obtained by performing DPI (Deep Packet Inspection) processing on ntopng streams. A total of 3,577,296 instances are collected, and each instance contains information of IP streams generated by network devices, namely source and destination IP addresses, ports, arrival interval, and Layer 7 protocols (applications) used as classes on the streams. This collection method ensures that the data can truly reflect the network activities in this network environment.

A key part of the data set is to label each network traffic record with a Protocol, and label the network protocol name corresponding to the traffic. These protocols cover many common network protocols, such as HTTP, FTP, TCP, UDP and so on. Protocol labeling provides a clear target label for subsequent model training, which enables the machine learning model to learn the mapping relationship between network traffic characteristics and protocol types, thus realizing accurate classification and prediction of unknown network traffic protocols. At the same time, through the analysis of the traffic characteristics of different protocols, it is also helpful to deeply understand the characteristics and behavior patterns of various network protocols and provide valuable reference for network security monitoring, network performance optimization and other fields.

3.2. Feature description

The data set contains many features, which describe the characteristics of network traffic from different dimensions and can be roughly divided into the following categories:

Time correlation feature: there is a Timestamp correlation feature, which records the time information of data packets in network traffic, such as the arrival time of data packets or the connection establishment time. This feature is very important for analyzing the changing trend of network traffic with time and capturing the time law of network activities. For example, we can judge the peak and low periods of network use by analyzing the characteristics of network traffic in different time periods.

Packet size characteristics: including the minimum length and average length of forward (Fwd) and backward packets. These characteristics reflect the change of packet size during transmission. Different network applications often have different packet size distribution patterns. For example, the packet size of file transfer protocol (FTP) may be different from that of real-time communication protocols (such as protocols used by instant messaging software). Analyzing these characteristics is helpful to identify the application protocols corresponding to network traffic.

Flag bit features: It involves various TCP flag bit counting features, such as FIN.Flag.Count, SYN.Flag.Count, PSH.Flag.Count, ACK.Flag.Count, URG.Flag.Count, etc. These flag bits play a key role in the process of TCP connection establishment, data transmission and connection closure. The combination and change of these flag bits have specific rules under different network connection states and application scenarios. For example, in the three-way handshake process of TCP connection establishment, SYN flag bits are used to initiate connection requests, and ACK flag bits are used to confirm connections, etc. By analyzing the characteristics of these flag bits, the state and behavior pattern of network connection can be inferred.

Statistical characteristics of traffic: including forward and backward average number of packets, average number of bytes, average batch rate and other characteristics. These characteristics describe the transmission rate and data volume of network traffic from a macro perspective, and can reflect the traffic intensity and activity of network applications. For example, video streaming applications usually have a high traffic rate and a large amount of data, while text transmission applications have a relatively low traffic rate. Through these characteristics, different types of network applications can be distinguished and identified.

3.3. Descriptive statistical analysis

Figure 1 visually shows the correlation between the characteristic variables in the data set through the color depth. The closer the color is to red, the stronger the positive correlation between variables; The closer the color is to blue, the stronger the negative correlation; A color close to white indicates a weak correlation. As can be seen from the figure, there is a significant correlation between some features. For example, some characteristics related to traffic statistics, such as forward average number of bytes and forward average number of packets, may show a positive correlation, which means that when the average number of packets increases in network traffic, the average number of bytes may also increase, reflecting the difference between the number of packets and the amount of data in some network applications when transmitting data. The correlation between some TCP flag bit features and other features is also worthy of attention. By analyzing these correlations, we can deeply understand the interaction between features in different network connection States, and provide reference for feature selection and combination in subsequent model construction.

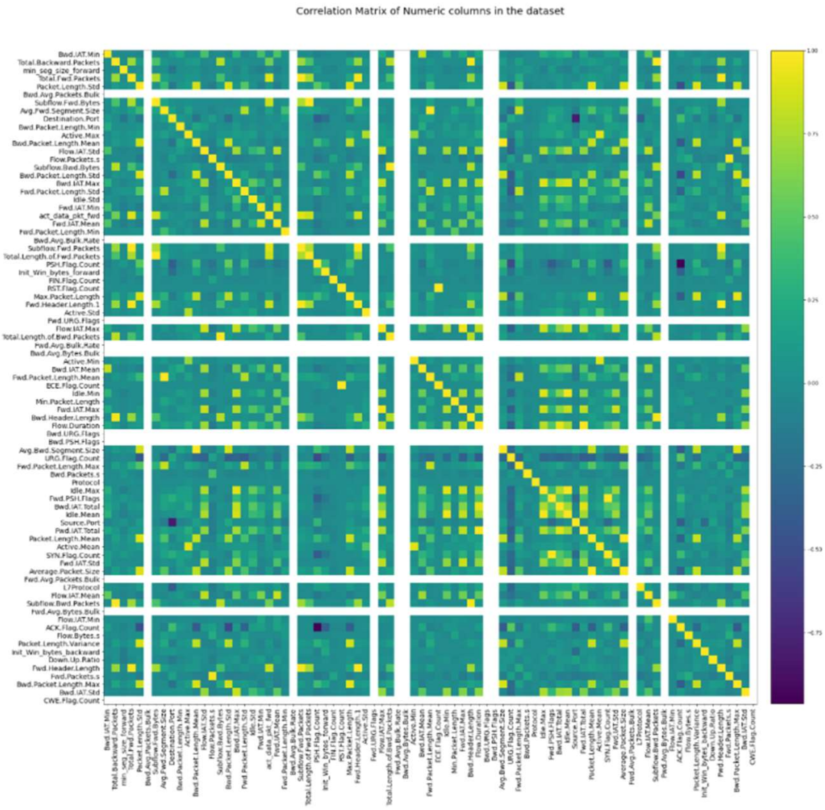

Figure 1. Characteristic variable correlation heat map.

Figure 2 shows the distribution of some variables, which is helpful to intuitively understand the characteristics of data concentration trend and dispersion degree. As can be seen from the figure, the distribution patterns of different variables are different. By analyzing the distribution of these variables, we can preliminarily judge the quality of data, identify possible abnormal values, provide

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 April 2025 doi:10.20944/preprints202504.1948.v1

important information for data preprocessing and subsequent model training, and ensure that the model can learn the characteristics and laws of network traffic data more accurately.

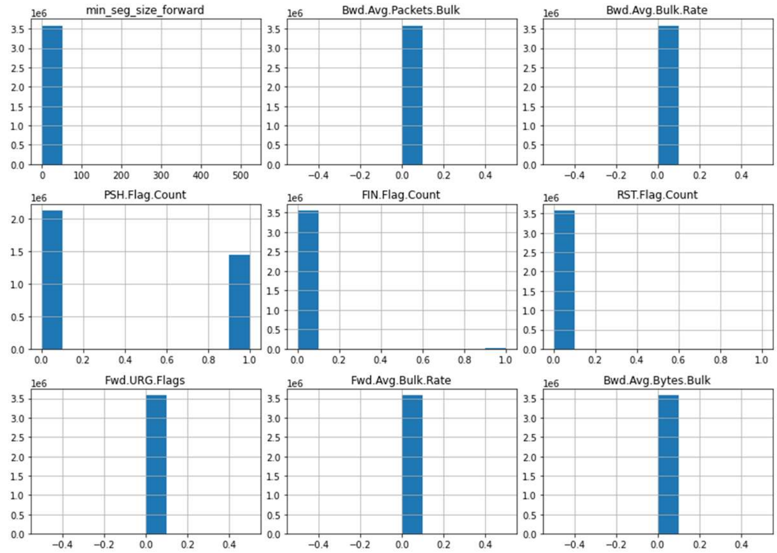

Figure 2. Descriptive distribution diagram of some variables.

4. Model Introduction

In the complex and key task of network traffic protocol classification, this study innovatively introduces CNN-LSTM model, aiming at giving full play to the respective advantages of Convolutional Neural Network, CNN) and Long Short-$ TERM Memory Network (LSTM), and realizing more accurate feature extraction and classification prediction of network traffic data.

As a feed-forward neural network, CNN is outstanding in processing data with grid structure, such as images, and its core advantages are local perception and weight sharing. In the scenario of network traffic analysis, the convolution layer of CNN can automatically learn the local characteristics of network traffic data, such as packet size characteristics, TCP flag bit characteristics and so on. Through the sliding convolution operation of different convolution kernels on data, various representative local patterns can be extracted, which are very important for identifying the traffic characteristics of different network protocol types. For example, a specific packet size distribution pattern or a flag bit combination pattern may be a unique identifier of a certain network protocol, and CNN can effectively capture these local characteristic information. At the same time, the pooling layer further reduces the dimension of the feature map output by the convolution layer, reduces the amount of data and calculation on the basis of retaining the key features, and improves the training efficiency and generalization ability of the model.

As an improved variant of Recurrent Neural Network, RNN, LSTM is mainly used to process data with time series characteristics, which effectively solves the problems of gradient disappearance and gradient explosion in traditional RNN. In network traffic data, time-related features contain abundant information. LSTM can make full use of these time series information to learn the changing trend and long-term dependence of network traffic with time. For example, through the analysis of network traffic at different time points, LSTM can capture the peak and trough periods of network usage and the activity rules of different protocols at different time periods. Its internal gating mechanism can flexibly control the inflow, outflow and memory of information, ensure that the model can selectively remember important time series information, and provide strong support for the classification of network traffic protocols.

In this study, CNN and LSTM are organically combined to construct CNN-LSTM model. Firstly, CNN convolution and pooling operations are used to extract the features of network traffic data, and the original high-dimensional traffic data is transformed into low-dimensional feature vectors with key features.

$$Y_{i,j} = \sum_{m,n} W_{m,n} \cdot X_{i+m,j+n} + b$$

$$Z_{i,j} = \max_{m,n} \ X_{i \times s+m,j \times s+n}$$

Where (i,j) represents the position of elements in the output feature graph and (m,n) represents the position of elements in the convolution kernel. And, s is the pool step size, which is usually greater than 1.

Then, these feature vectors are input into the LSTM network. Based on its time series processing ability, LSTM further analyzes and learns these features, and excavates the time dependence between features.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

Where f_t is the output of the forgetting gate at the moment of t, σ is the Sigmoid activation function, W_f is the weight matrix of the forgetting gate, $[h_{t-1}, x_t]$is the splicing of the hidden state at the last moment and the input at the current moment, and b_f is the offset term. The output value of forgetting gate is between 0 and 1, where 1 means to keep all information and 0 means to discard all information. In the analysis of network traffic, the forgetting gate can decide whether to keep the information such as the network activity time law memorized before according to the traffic characteristics of the current and last moment.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\widetilde{C_t} = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

Where i_t is the output of the input gate, $\widetilde{C_t}$ is the candidate cell state generated at the current moment, W_i and W_c are the corresponding weight matrices, b_i and b_c are the bias terms, tanhis the hyperbolic tangent activation function. The output of the input gate controls the proportion of new information entering the cell state, and the candidate cell state $\widetilde{C_t}$ contains the new information extracted at the current moment.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C_t}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \cdot \tanh(C_t)$$

Where o_t is that output of the output gate and h_t is the hidden state of the current moment. The output of the output gate controls which information in the cell state will be output, and the cell state processed by hyperbolic tangent function is multiplied by the output of the output gate to get the hidden state h_t at the current moment as the final output of the LSTM unit. In the network traffic prediction, the hidden state h_t can be used to predict the network traffic protocol type and other information at the next moment.

In this way, CNN-LSTM model can take into account the local characteristics and time series characteristics of network traffic data at the same time. Compared with a single CNN or LSTM model, CNN-LSTM model has stronger feature learning and pattern recognition capabilities, thus realizing the classification and prediction of unknown network traffic protocols more accurately, and providing a more efficient and accurate model solution for the field of network traffic analysis.

5. Model results analysis

When evaluating different models and algorithms, this study analyzes them from three dimensions: accuracy, F1 score and the number of features. Relevant data are presented in tables, and the specific analysis is as follows:

In Table 1, the accuracy of different models under different algorithms is significantly different. CNN-LSTM model is the most outstanding, and its accuracy is as high as 0.981 when feature selection algorithm is used. This shows that the model can effectively use the filtered features to accurately identify the protocol types of network traffic. In contrast, the accuracy of LSTM and CNN models under the same algorithm is low, which is 0.793 and 0.465, respectively, which shows that CNN-LSTM model has stronger feature learning and pattern recognition ability when dealing with this

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 April 2025 doi:10.20944/preprints202504.1948.v1

kind of network traffic data. When principal component analysis (PCA) algorithm is used, the accuracy of the three models is greatly reduced, which is 0.687 for CNN-LSTM model, 0.299 for LSTM model and 0.201 for CNN model. This may be because the PCA algorithm loses some information that is crucial to model classification in the process of dimensionality reduction, thus affecting the model performance. Under the K-folding algorithm, the accuracy of CNN-LSTM model is 0.945 and CNN model is 0.464, which are relatively stable, but the accuracy of LSTM model is only 0.183, which shows that the generalization ability of LSTM model is relatively weak under this algorithm. When the Original feature is used, the accuracy of CNN-LSTM model is still high (0.956), while the accuracy of LSTM and CNN model is 0.181 and 0.467 respectively, which proves that CNN-LSTM model has better adaptability to the original feature.

Table 1. Accuracy for each model for particular algorithm.

	CNN-LSTM	LSTM	CNN
Feature Selection	**0.981**	0.793	0.465
PCA	**0.687**	0.299	0.201
K-Cross Folding	**0.945**	0.183	0.464
Original	**0.956**	0.181	0.467

The F1 score comprehensively considers the accuracy and recall of the model. From Table 2, under the feature selection algorithm, the F1 scores of CNN-LSTM, LSTM and CNN models are close, which are 0.063, 0.060 and 0.058 respectively. This means that there is little difference between the three models in balance accuracy and recall after screening features. Under PCA algorithm, the F1 scores of the three models are similar, ranging from 0.057 to 0.060, indicating that PCA algorithm has little influence on the F1 scores of the models. However, under the K-fold cross-validation algorithm, the F1 score of CNN-LSTM model is 0.945, which is in sharp contrast with 0.183 of LSTM model and 0.464 of CNN model, which further proves that CNN-LSTM model has better comprehensive performance under this algorithm. Under the condition of original features, the F1 score of CNN-LSTM model is 0.956, which is much higher than that of LSTM and CNN models, once again showing the advantages of this model in processing original data.

Table 2. F1 Scores for each model for particular algorithm.

	CNN-LSTM	LSTM	CNN
Feature Selection	**0.063**	0.060	0.058
PCA	0.057	**0.060**	0.058
K-Cross Folding	**0.945**	0.183	0.464
Original	**0.956**	0.060	0.058

As can be seen from Table 3, when the feature selection algorithm is adopted, the number of features of CNN-LSTM, LSTM and CNN models is 29; When PCA algorithm is used, the number of features is reduced to 13; However, in the case of K-fold cross-validation algorithm and original features, the number of features is 69. Combining the analysis results of accuracy and F1 score, CNN-LSTM model can still maintain high performance when the number of features is reduced (such as feature selection or PCA algorithm), which shows that the model has good adaptability to the change of feature number. However, the performance of LSTM and CNN models fluctuates greatly when the number of features changes, especially LSTM model. When the number of features is large (such as original features), the performance drops obviously, which shows that the sensitivity of different models to the number of features is different.

Table 3. Number of Features for each model and algorithm.

	CNN-LSTM	LSTM	CNN
Feature Selection	29	29	29

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 April 2025 doi:10.20944/preprints202504.1948.v1

PCA	13	13	13
K-Cross Folding	69	69	69
Original	69	69	69

6. Conclusions

This study focuses on network traffic data, and has done systematic work in data collection, feature analysis, model construction and evaluation, and achieved a series of valuable results.

In model research, the performances of CNN-LSTM, LSTM and CNN under different algorithms are comprehensively evaluated. The results show that CNN-LSTM model has the best overall performance in terms of accuracy and F1 score. When using the feature selection algorithm, the accuracy of CNN-LSTM model reaches 0.981, which shows strong feature learning and pattern recognition ability, and can effectively identify the network traffic protocol type by using the filtered features. When dealing with the original features, the model can also maintain high accuracy (0.956) and F1 score (0.956), and has good adaptability to the original data. In contrast, LSTM and CNN models are relatively weak in performance and more sensitive to the change of feature number, especially LSTM model, which has poor generalization ability under some algorithms.

On the whole, this study verifies the effectiveness and superiority of CNN-LSTM model in the task of network traffic protocol classification. At the same time, the analysis of data characteristics and the comparison of different models and algorithms in the research process provide valuable experience for the subsequent research on network traffic analysis. Future research can further explore more optimized model structure and feature processing methods to cope with more complex network environment and traffic data, promote the development of network traffic analysis, and provide more powerful support for practical applications such as network security monitoring and network performance optimization.

References

1. Azab A, Khasawneh M, Alrabaee S, et al. Network traffic classification: Techniques, datasets, and challenges[J]. Digital Communications and Networks, 2024, 10(3): 676-692.
2. Shafiq M, Yu X, Laghari A A, et al. Network traffic classification techniques and comparative analysis using machine learning algorithms[C]//2016 2nd IEEE International Conference on Computer and Communications (ICCC). IEEE, 2016: 2451-2455.
3. Salman O, Elhajj I H, Kayssi A, et al. A review on machine learning–based approaches for Internet traffic classification[J]. Annals of Telecommunications, 2020, 75(11): 673-710.
4. Patel S, Gupta A, Kumari S, et al. Network traffic classification analysis using machine learning algorithms[C]//2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN). IEEE, 2018: 1182-1187.
5. Piskac P, Novotny J. Network traffic classification based on time characteristics analysis[J]. Masaryk University Faculty of Informatics, 2011: 4-14.
6. Kumar R, Swarnkar M, Singal G, et al. IoT network traffic classification using machine learning algorithms: An experimental analysis[J]. IEEE Internet of Things Journal, 2021, 9(2): 989-1008.
7. Soysal M, Schmidt E G. Machine learning algorithms for accurate flow-based network traffic classification: Evaluation and comparison[J]. Performance Evaluation, 2010, 67(6): 451-467.
8. Dias K L, Pongelupe M A, Caminhas W M, et al. An innovative approach for real-time network traffic classification[J]. Computer networks, 2019, 158: 143-157.
9. Hwang R H, Peng M C, Nguyen V L, et al. An LSTM-based deep learning approach for classifying malicious traffic at the packet level[J]. Applied Sciences, 2019, 9(16): 3414.
10. Camelo M, Soto P, Latré S. A general approach for traffic classification in wireless networks using deep learning[J]. IEEE Transactions on Network and Service Management, 2021, 19(4): 5044-5063.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 April 2025 doi:10.20944/preprints202504.1948.v1

11. Nithya B, Venkataraman V, Nithin Balaaji D V, et al. A CNN-LSTM Approach for Classification of Major TCP Congestion Control Algorithms[C]//Intelligent Sustainable Systems: Selected Papers of WorldS4 2021, Volume 2. Springer Singapore, 2022: 579-591.

12. Lopez-Martin M, Carro B, Sanchez-Esguevillas A, et al. Network traffic classifier with convolutional and recurrent neural networks for Internet of Things[J]. IEEE access, 2017, 5: 18042-18050.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.