Lightweight Multi-Input Shape CNN-based Application Traffic Classification

Ui-Jun Baek, Min-Seong Lee, Jee-Tae Park, Jeong-Woo, Choi Chang-Yui Shin, Ju-Sung Kim, Yoon-Seong Jang and Myung-Sup Kim
Dept. of Computer Information Science, Korea University
Sejong, Korea
{pb1069, min0764, pjj5846, choigoya97, realmine, jsung0514, brave1094, tmskim}@korea.ac.kr

***Abstract*— This research focuses on the input shape of CNN-based application traffic. The previously proposed multi-input model CNN classification method classified applications through various shapes of features derived from fixed-length packets, achieving a higher classification accuracy compared to traditional CNNs. However, it had limitations such as vulnerability to overfitting despite its high classification accuracy and slow inference speed. To overcome these challenges, we introduce a lightweight version of the previously proposed MISCNN, called MISCNN+. MISCNN+ demonstrated approximately 2.9 times faster inference speed and a 3.6% improvement in classification accuracy compared to the previous version.**

Keywords—Application Traffic Classification, CNN, Input-Shape, Lightweight

I. INTRODUCTION

In recent years, due to the proliferation of various applications and services, the field of application traffic classification, which is a subdomain of network management, has been gaining increasing importance. Application traffic classification is the process of categorizing network traffic based on the applications or services responsible for generating it. Accurate application traffic classification is crucial for understanding network traffic patterns and optimizing network performance. Network administrators can make informed decisions about resource allocation and capacity planning by identifying specific applications through traffic classification. Application traffic classification finds applications in various domains, including network optimization, network security, quality of service (QoS) assurance, regulatory compliance, billing, and accounting.

The technology of application traffic classification has evolved over time, transitioning from traditional heuristic methods to current deep learning-based classification techniques. Notably, there has been extensive research into CNN-based application traffic classification, with impressive research outcomes. However, prior studies in CNN-based application traffic classification did not fully consider the shape of the input model for CNNs. Depending on the shape of the input model and kernel, CNNs can extract features of varying shapes. Recognizing this, Baek et al. proposed the MISCNN method, which simultaneously learns different shapes derived from fixed-length byte sequences. This approach achieved a higher classification accuracy compared to conventional studies using simple 1D or square-shaped models. However, MISCNN had limitations such as high computational resource consumption and slow inference speed.

In this paper, we introduce MISCNN+, a lightweight version of the existing MISCNN model. MISCNN+ leverages Global Average Pooling (GAP) and Global Max Pooling (GMP) to extract more generalized features compared to the original model, resulting in a 3.6% improvement in classification accuracy. Furthermore, by selectively adopting and applying only a subset of shapes derived from fixed-length byte sequences, we significantly reduce computational resource consumption. MISCNN+ demonstrates approximately 2.9 times faster inference speed compared to the previous version.

The remaining sections of the paper cover related research, dataset descriptions, the proposed methodology, experimental results, and conclusions.

II. RELATED WORKS

TABLE I. LIST OF INPUT SHAPES USED IN RECENT STUDIES

#	Year	Input shape	Dimension	Input size
[1]	2021	Linear	1D	784
[10]	2021	Linear	1D	1480
[8]	2021	Square	2D	(28, 28)
[6]	2021	Linear	1D	256
[7]	2022	Linear	1D	not provided
[11]	2022	Linear	1D	1480
[5]	2022	Linear	1D	1480
[9]	2022	Square	2D	(28, 28)
[13]	2022	Linear	1D	1500

In this chapter, we investigate the latest CNN-based studies using the ISCX VPN-non VPN 2016 dataset and focus on the input shape used in these studies. Table 1 presents the shape and size of the input used in recent CNN-based application traffic classification studies. It is evident that the majority of studies have adopted a linear shape, with approximately half of them setting the input size close to the MTU (Max Transmission Unit). According to comparative results from some studies, it has been reported that 1D CNNs using a linear shape outperform 2D CNNs using a square shape [12, 13]. Furthermore, most studies do not provide criteria for determining the shape and size of the input. Some studies have indicated the following criteria for determining the shape or size: [1] set the shape and size to facilitate comparison with previous research, while [8] set the input shape to match the input shape of the deep learning backbone network. [13] compared the classification results of 1D CNNs using a linear shape and 2D CNNs using a square shape. Most studies do not consider the input shape or provide criteria for selecting the respective shape. However, we have experimentally demonstrated that if an appropriate input

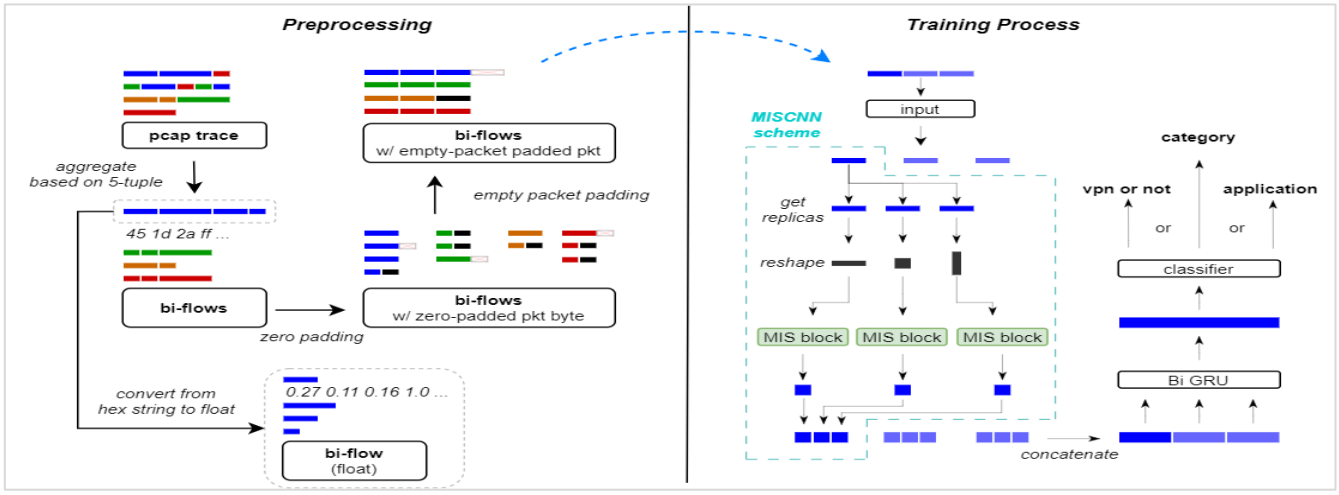

Fig. 1. Overview of MISCNN+

shape is chosen based on prior research, it can enhance classification performance [2].

III. Preliminaries

A. Dataset description

The ISCX VPN-nonVPN 2016 dataset in raw pcap format is used to evaluate the classification performance [3]. This dataset includes traffic from various applications, covering normal sessions and sessions encapsulated over VPN. It serves as the basis for evaluating the model's performance in three tasks. Firstly, the model is trained to classify whether the input traffic was encapsulated over VPN or not (binary classification). Secondly, it categorizes the input traffic into six categories based on similar application functions. For example, Netflix and YouTube are grouped under the "Streaming" category. Lastly, the model categorizes the input traffic based on the application that generated it. This paper specifically focuses on category classification among the three tasks. Detailed information about these tasks is provided in Table 2.

TABLE II. CLASSIFICATION TASK OF ISCX VPN-NONVPN 2016

Tasks	Classes
Encapsulation	VPN, non-VPN
Category	VoIP, File Transfer, P2P, Streaming, Chat, Email
Application	Skype, Torrent, Hangouts, VoIP Buster, Facebook, FTPs, SCP, Email, Youtube, Vimeo, Spotify, Neflix, SFTP, Aim, ICQ

B. Dataset cleaning

To evaluate the proposed method, the dataset with a PCAP extension is preprocessed and fed into a deep learning model as input. In order to ensure the effectiveness of the learning process, all packets in the dataset are reconstructed into bidirectional flows based on the same 5-tuple. However, it should be noted that a portion of these reconstructed bidirectional flows may not be relevant to the actual application functions, posing a challenge for the learning process. In addition, incomplete TCP flows (lacking the 3-way handshake) were excluded from the dataset to avoid any potential interference with the learning of temporal features in the packet flow. Following the preprocessing step, a total of 29.4K bidirectional flows were successfully extracted.

IV. Proposed Method

A. Overview of MISCNN

The training of the cleaned dataset to create an inference model involves two main steps: preprocessing and training, as depicted in Figure 1. In the preprocessing step, the input PCAP traces are preprocessed to conform to the input requirements of the training model. The PCAP traces are aggregated into bi-flows based on the 5-tuple of each packet, and the packet bytes are converted into real numbers. Given that the input to the training model needs to be normalized, we either zero-pad the packet byte sequence or truncate the bytes to match a predetermined size. Moreover, the number of packets within a flow can vary depending on the application's behavior. To handle this, we add or remove empty packets to align the flow with the predetermined number of packets. In the training process, the preprocessed three-dimensional data is fed into the model on an instance-by-instance basis. Depending on the batch size, multiple instances can be simultaneously input and trained. Each instance represents a single bi-flow, and the MISCNN scheme extracts multifaceted features from each bi-flow. To accomplish this, each packet within the input bi-flow is duplicated and reshaped into different shapes. The reshaped packets then pass through MIS blocks, which are tailored for each input shape, to extract spatial features from the packets. The extracted spatial features from each packet are concatenated and processed through a bi-GRU (bidirectional Gated Recurrent Unit), which captures the temporal features of the packets within the flow and linearly transforms them into features for final classification. These transformed features, acquired through the learning process, are employed for the three classification tasks: encapsulation, category, and application.

Within the MISCNN scheme, the MIS block plays a crucial role in extracting spatial features from packets with varying shapes. Comprised of six residual blocks [4], as illustrated in (a) of Figure 2, the MIS block progressively increases the number of channels. Beginning with eight channels, the block doubles the channel count, eventually reaching 64 channels. To manage the abundance of features and prevent overfitting, the extracted spatial features are condensed and combined through Global Average Pooling (GAP), Global Max Pooling (GMP), and a Fully-connected (FC) layer.

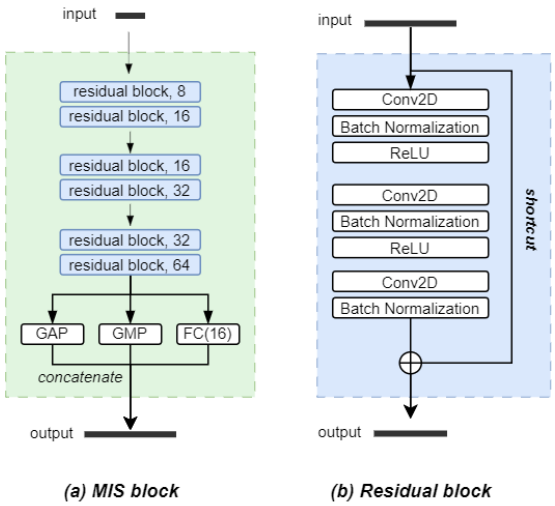

Fig. 2.Overview of MIS block

B. *Ligthening*

There are two methods employed to enhance the lightweight nature of the proposed approach.

The first method is to apply Global Average Pooling (GAP) and Global Max Pooling (GMP) to the spatial features obtained after passing through the MIS block. Global Average Pooling calculates the average value of each feature map across all spatial locations, providing a feature vector that summarizes the information. Similarly, Global Max Pooling selects the maximum value from each feature map at every spatial location, generating a feature vector that captures the most important information. These pooling techniques help prevent overfitting, reduce computation, and are less sensitive to small spatial transformations compared to other pooling methods like max pooling. Additionally, the utilization of Fully-Connected layers (FCs) alongside GAP and GMP is recommended to mitigate information loss. By incorporating small FCs, the approach can achieve significant accuracy improvements without a significant increase in computational cost.

The second method involves selectively using a subset of shapes in a multi-shape split of the MISCNN scheme. The MISCNN method employs filters that extract features from multiple models based on a single packet, allowing for a detailed representation of the flow. However, overly detailed features can lead to overfitting and increased inference time. To address this, it is suggested to select only a subset of models extracted from a single packet. For instance, when the packet size is 784, the number of extractable models can be 15, which aligns with the number of divisors of the packet size. Table 3 not only presents the available models that can be extracted from a single packet but also outlines the configurations of the Kernel and Strides for the Convolution layers applied to each model.

V. EXPERIMENTS

A. *Metrics setup*

The main evaluation metrics used with the proposed methodology are Accuracy, F-measure, which are commonly employed in the field of network traffic classification. Additionally, in some evaluations, assessment is conducted regarding the inference time.

TABLE III. CLASSIFICATION TASK OF ISCX VPN-NONVPN 2016

Input shape	Kernel	Strides
(1, 784)	(1, 98)	(1, 2)
(2, 392)	(1, 49)	(1, 2)
(4, 196)	(1, 24)	(1, 2)
(7, 112)	(1, 14)	(1, 2)
(8, 98)	(1, 12)	(1, 2)
(14, 56)	(1, 7)	(1, 2)
(16, 49)	(2, 6)	(2, 2)
(28, 28)	(3, 3)	(2, 2)
(49, 16)	(6, 2)	(2, 2)
(56, 14)	(7, 1)	(2, 1)
(98, 8)	(12, 1)	(2, 1)
(112, 7)	(14, 1)	(2, 1)
(196, 4)	(24, 1)	(2, 1)
(392, 2)	(49, 1)	(2, 1)
(784, 1)	(98, 1)	(2, 1)

B. *Influence of shape combination*

In this paragraph, we present a comparison of classification accuracy and inference time based on different combinations of shapes. These combinations consist of up to three models, denoted by the central index "m" and the last index "z" in the shape array. To conduct the comparison, we generated a set of 16 random shape combinations, including 2 combinations using a single shape (1D or 2D CNN), 8 combinations using 2 shapes, and 6 combinations using 3 shapes. The 17th combination at the bottom represents the scenario where all shapes are used. Table 4 provides a comparison of Accuracy, F1-score, and Inference Time for these 17 combinations.

Based on the experimental results, the model using the 8th combination exhibited the highest performance, demonstrating approximately 6-7% improvement in Accuracy and F1-score compared to the first combination (1D-CNN) and the second combination (2D-CNN). Additionally, it showed a 1.3% improvement compared to the third combination that combined 1D-CNN and 2D-CNN from a previous study [11], as well as approximately 3.5% improvement compared to the MISCNN baseline that utilized all shapes. In terms of inference time, the model using the 8th combination enabled the classification of 43% fewer flows compared to the model based on 2D-CNN within the same time. However, considering that classification accuracy is generally more important than inference time in the field of traffic classification, the trade-off between accuracy and inference time is justified. The 7% improvement in accuracy and 43% decrease in inference time (when comparing combination 8 with combination 2) for the inference model are reasonable. Furthermore, when comparing the 8th combination with the MISCNN baseline, there is a 3.4% improvement in accuracy and a significant 191% improvement in inference time.

C. *Influence of applying the GMP layer and GAP layer*

Table 5 illustrates the impact of incorporating the GMP/GAP layer on the performance of MISCNN+. By combining the results from a FC (fully connected) layer with 16 units, GMP layer, and GAP layer, we achieve an accuracy improvement of 3.6% and an f-measure improvement of 3.3% compared to using only the fully connected layer with 32 units.

TABLE IV. EFFECT OF SHAPE COMBINATION ON THE PERFORMANCE OF THE MISCNN+

#	1st shape	2nd shape	3rd shape	Accuracy	F1-score	Inference time (flows/s)
1	0			0.906	0.908	1033
2	m			0.901	0.901	1415
3	0	m		0.959	0.959	803
4	0	m+1		0.916	0.917	767
5	1	m		0.961	0.962	796
6	1	m+1		0.919	0.921	769
7	1	m+2		0.916	0.918	693
8	**2**	**m**		**0.972**	**0.97**	**808**
9	2	m+1		0.934	0.938	707
10	2	m+2		0.918	0.919	769
11	0	m+1	z-1	0.924	0.925	627
12	0	m	z-2	0.956	0.956	606
13	1	m+1	z-1	0.928	0.929	609
14	1	m	z-2	0.926	0.962	654
15	0	m+1	z-1	0.923	0.924	617
16	0	m	z-2	0.958	0.958	649
17	MISCNN (all shape used)			0.938	0.934	277

TABLE V. THE IMPACT OF APPLYING THE GMP/GAP LAYER ON THE PERFORMANCE OF MISCNN+

	Accuracy	F1-score
FC(16) + GMP + GAP	0.972	0.97
FC(32)	0.936	0.937

D. Influence of applying the GMP layer and GAP layer

Table 6 presents the performance comparison results between MISCNN+ and existing research methods in the field of application traffic classification. MISCNN+ achieves a 3.5% higher accuracy and a 5.5% higher F-measure compared to the existing research methods.

TABLE VI. COMPARISON WITH OTHER METHODS

Method	Ref. no	Accuracy	F-measure
1D-CNN	[14]	0.874	0.835
2D-CNN	[15]	0.874	0.835
1D-CNN	[16]	0.829	0.762
DISTILLER	[1]	0.937	0.915
MISCNN	[2]	0.938	0.934
MISCNN++ (proposed)	-	0.972	0.97

VI. CONCLUSION

This paper introduces the improved Multi Input Shape CNN (MISCNN+) method, which can be universally applied to CNN-based application traffic classifiers, and proposes two enhancements to improve its performance. Existing CNN-based models for application traffic classification accept only one-dimensional or square input vectors, limiting the flexibility of input shapes. To address this, we propose MISCNN, which combines multiple models derived from fixed-length packets to extract features. However, increasing the number of shapes can lead to slower inference speed and overfitting. To overcome these limitations, we propose two enhancements.

Firstly, we utilize Global Average Pooling (GAP) and Global Max Pooling (GMP) to enhance the generalization capability of the models. The models with GAP and GMP applied achieve a 3.6% improvement in accuracy and a 3.3% improvement in F-measure compared to the method that aggregates features into the fully connected layer from the conventional Residual block. Secondly, instead of using all possible shapes derived from fixed-length packets, we select a subset of shapes. For instance, if the input packet size is 784 bytes, there are 15 possible shapes that can be derived, corresponding to the number of divisors of the packet size. Using all these shapes incurs significant penalties in terms of inference time and tends to lead to overfitting. Therefore, in this paper, we aim to improve accuracy and inference time by selecting only a subset of derived shapes. Experimental results show that using a minimum of two shapes significantly improves accuracy, resulting in a 6-7% accuracy increase compared to the previous studies that used single one-dimensional or square vector inputs (1D-CNN-based, 2D-CNN-based). Additionally, the implemented ResNet-based backbone network and the inference model with the proposed method achieve comparable classification results to state-of-the-art research, and they can be applied to various model structures proposed in other CNN-based classification studies without constraints.

Considering the limitations of the proposed method, several future research directions or challenges can be identified. Firstly, exploring advanced techniques to optimize deep learning models is a promising avenue. Secondly, improving the performance of application traffic classification in environments where IP headers are not available is a significant challenge. Our private experimental results demonstrate that the accuracy can decrease by up to 15% when the dataset trained without the IP header is used, indicating the model's reliance on information within the IP layer, particularly server-side addresses. To address this, methods that actively utilize payload information in the application layer instead of the IP layer can be explored, such as extracting new features (statistical features, etc.) or incorporating recent deep learning techniques (attention mechanisms, etc.).

ACKNOWLEDGEMENTS

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2023-00230661, Development of Standards for Hybrid Quantum Key Distribution Method and Network Management Technology) and was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) funded by the Korea government (00235509, Development of security monitoring technology based network behavior against encrypted cyber threats in ICT convergence environment).

REFERENCES

[1] Aceto, G., Ciuonzo, D., Montieri, A., Pescapé, A., 2021. DISTILLER: Encrypted traffic classification via multimodal multitask deep learning. Journal of Network and Computer Applications 183–184, 102985. https://doi.org/10.1016/j.jnca.2021.102985

[2] Baek, U., Kim, B., Park, J., Choi, J., Kim, M., 2022. MISCNN: A Novel Learning Scheme for CNN-Based Network Traffic Classification, in: 2022 23rd Asia-Pacific Network Operations and Management Symposium (APNOMS). Presented at the 2022 23rd Asia-Pacific Network Operations and Management Symposium (APNOMS), pp. 01–06. https://doi.org/10.23919/APNOMS56106.2022.9919961

[3] Draper-Gil, G., Lashkari, A.H., Mamun, M.S.I., A. Ghorbani, A., 2016. Characterization of Encrypted and VPN Traffic using Time-related Features:, in: Proceedings of the 2nd International Conference on Information Systems Security and Privacy. Presented at the 2nd International Conference on Information Systems Security and Privacy,

SCITEPRESS - Science and and Technology Publications, Rome, Italy, pp. 407–414. https://doi.org/10.5220/0005740704070414

[4] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90

[5] He, Y., Li, W., 2022. A Novel Lightweight Anonymous Proxy Traffic Detection Method Based on Spatio-Temporal Features. Sensors 22, 4216. https://doi.org/10.3390/s22114216

[6] Hu, X., Gu, C., Chen, Y., Wei, F., 2021a. CBD: A Deep-Learning-Based Scheme for Encrypted Traffic Classification with a General Pre-Training Method. Sensors 21, 8231. https://doi.org/10.3390/s21248231

[7] Izadi, S., Ahmadi, M., Nikbazm, R., 2022. Network traffic classification using convolutional neural network and ant-lion optimization. Computers and Electrical Engineering 101, 108024. https://doi.org/10.1016/j.compeleceng.2022.108024

[8] Lu, B., Luktarhan, N., Ding, C., Zhang, W., 2021. ICLSTM: Encrypted Traffic Service Identification Based on Inception-LSTM Neural Network. Symmetry 13, 1080. https://doi.org/10.3390/sym13061080

[9] Pathmaperuma, M.H., Rahulamathavan, Y., Dogan, S., Kondoz, A., 2022. CNN for User Activity Detection Using Encrypted In-App Mobile Data. Future Internet 14, 67. https://doi.org/10.3390/fi14020067

[10] Soleymanpour, S., Sadr, H., Nazari Soleimandarabi, M., 2021. CSCNN: Cost-Sensitive Convolutional Neural Network for Encrypted Traffic Classification. Neural Process Lett 53, 3497–3523. https://doi.org/10.1007/s11063-021-10534-6

[11] Telikani, A., Gandomi, A.H., Choo, K.-K.R., Shen, J., 2022. A Cost-Sensitive Deep Learning-Based Approach for Network Traffic Classification. IEEE Transactions on Network and Service Management 19, 661–670. https://doi.org/10.1109/TNSM.2021.3112283

[12] Xu, L., Zhou, X., Ren, Y., Qin, Y., 2019. A Traffic Classification Method Based on Packet Transport Layer Payload by Ensemble Learning, in: 2019 IEEE Symposium on Computers and Communications (ISCC). Presented at the 2019 IEEE Symposium on Computers and Communications (ISCC), pp. 1–6. https://doi.org/10.1109/ISCC47284.2019.8969702

[13] Zheng, W., Zhong, J., Zhang, Q., Zhao, G., 2022. MTT: an efficient model for encrypted network traffic classification using multi-task transformer. Appl Intell 52, 10741–10756. https://doi.org/10.1007/s10489-021-03032-8

[14] Wang, W., Zhu, M., Wang, J., Zeng, X., Yang, Z., 2017. End-to-end encrypted traffic classification with one-dimensional convolution neural networks, in: 2017 IEEE International Conference on Intelligence and Security Informatics (ISI). Presented at the 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 43–48. https://doi.org/10.1109/ISI.2017.8004872

[15] Huang, H., Deng, H., Chen, J., Han, L., Wang, W., 2018. Automatic Multi-task Learning System for Abnormal Network Traffic Detection. Int. J. Emerg. Technol. Learn. 13, 4. https://doi.org/10.3991/ijet.v13i04.8466

[16] Rezaei, S., Liu, X., 2020. Multitask Learning for Network Traffic Classification, in: 2020 29th International Conference on Computer Communications and Networks (ICCCN). Presented at the 2020 29th International Conference on Computer Communications and Networks (ICCCN), pp. 1–9. https://doi.org/10.1109/ICCCN49398.2020.9209652