Preprocessing and Analysis of an Open Dataset in Application Traffic Classification

Ui-Jun Baek
Computer Information Science
Korea University
Sejong, Korea
pb1069@korea.ac.kr

Min-Seong Lee
Computer Information Science
Korea University
Sejong, Korea
min0764@korea.ac.kr

Jee-Tae Park
Computer Information Science
Korea University
Sejong, Korea
pjj5846@korea.ac.kr

Jeong-Woo Choi
Computer Information Science
Korea University
Sejong, Korea
choigoya97@korea.ac.kr

Chang-Yui Shin
Defense Agency for
Technology and Quality
Daejeon, Korea
superego99@dtaq.re.kr

Myung-Sup Kim
Computer Information Science
Korea University
Sejong, Korea
tmskim@korea.ac.kr

Abstract— **Data preprocessing is a crucial step in data analysis and machine learning. This step involves transforming raw data into a suitable format for analysis, removing noise, and handling outliers to improve data quality. In particular, it offers benefits such as providing accurate analysis results, enhancing model performance, improving model generalization capabilities, and enabling faster model training. The public dataset (ISCX VPN-nonVPN 2016) is widely used in the field of application traffic classification. However, each study may have different methods for preprocessing the dataset, and detailed explanations or publicly available preprocessed datasets are not provided. Therefore, objective performance evaluation between methodologies becomes challenging. This paper performs preprocessing on the widely used public dataset (ISCX VPN-nonVPN 2016) in the field of application traffic classification and analyzes it. Additionally, it releases the preprocessed dataset publicly, enabling objective performance comparisons with other papers that utilize this public dataset.**

Keywords— ***Application Traffic classification, Preprocessing, ISCX VPN-nonVPN 2016, Open Dataset, Deep Learning***

I. INTRODUCTION[1]

Data preprocessing is a critical step in data analysis and machine learning tasks. This step involves transforming raw data into a suitable format for analysis, removing noise, and handling missing values or outliers to enhance data quality. The importance of data preprocessing can be explained for the following reasons:

- **Accurate analysis results:** Tasks such as noise removal, outlier handling, and missing value imputation reduce data distortions and enable the derivation of reliable outcomes.
- **Enhanced model performance:** Data often contains outliers or missing values, and applying such data directly to models can lead to degraded prediction performance. Proper treatment of outliers and missing values through data preprocessing enables models to make more accurate predictions and generalizations.
- **Improved model generalization capability:** It may include tasks such as resizing data, performing feature scaling, and encoding categorical data. These preprocessing steps enable models to learn generalized patterns across diverse data.
- **Faster analysis and model training:** As data preprocessing is performed as part of the analysis workflow, improving data quality and transforming it into a suitable format for models can shorten the time needed for analysis and model training.

The open dataset (ISCX VPN-nonVPN 2016) is widely used in the field of application traffic classification [1], but each study has different preprocessing methods for it. Additionally, these papers either provide detailed explanations of the preprocessing methods or do not disclose the preprocessed dataset to ensure reproducibility. Considering the significant variation in results based on the preprocessing methods and the division of the preprocessed dataset into training, validation, and test sets in the machine learning field, it is desirable for each study to provide clear preprocessing criteria and share the preprocessed dataset for comparison with other studies. This paper performs preprocessing on the widely used open dataset (ISCX VPN-nonVPN 2016) in the field of application traffic classification and analyzes it. The proposed preprocessing method considers the behavior of applications and preprocesses the data based on protocols, which is explained in detail. Additionally, it improves the performance of application traffic classification by removing TCP flows with missing TCP 3-way handshake. The results of classifying the dataset with the proposed method using known deep learning models validate the soundness of the proposed approach. The paper then presents related studies that use the same open dataset in the introduction and compares them. Subsequently, it describes the detailed specifications and preprocessing steps of the open dataset. In the experimental section, it classifies application traffic using known deep learning models and analyzes the experimental results. Finally, the paper discusses the contributions, limitations, and future research directions of this study.

II. RELATED WORKS

[2] proposed a novel multimodal multitask deep learning approach called the DISTILLER classifier for traffic classification.

[1] This work was supported by the Technology Innovation Program grant funded By the Ministry of Trade, Industry & Energy (MOTIE, Korea) and the Korea Evaluation Institute of Industrial Technology (KEIT) (No. 20008902, Development of SaaS SW Management Platform based on 5Channel Discovery technology for IT Cost Saving) and "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) (2021RIS-004).

TABLE I. SPECIFICATIONS OF THE ORIGINAL DATASET AND PREPROCESSED DATASET WITH EACH PREPROCESSING METHOD APPLIED.

Category	Original			By Protocol			By 3-handshake		
	Total Packet Size (MB)	#Packet (K)	#Flow (K)	Total Packet Size (MB)	#Packet (K)	#Packet (K)	Total Packet Size (MB)	#Packet (K)	#Packet (K)
Chat	62	234.49	13.96	54	154.53	2.56	27	126.04	13.74
	0.78 0.22	0.59 0.41	0.04 0.96	0.9 0.1	0.9 0.1	0.2 0.8	0.49 0.51	0.24 0.76	0.02 0.98
Email	17	78.74	8.06	15	52.55	1.84	9	53.03	7.99
	0.7 0.3	0.57 0.43	0.06 0.94	0.82 0.18	0.85 0.15	0.25 0.75	0.39 0.61	0.36 0.64	0.05 0.95
File Transfer	17,827	11,310	70.3	10,926	8534	7.59	15,414	10141	70.01
	0.95 0.05	0.88 0.12	0.03 0.97	1 0	1 0	0.26 0.74	0.95 0.05	0.86 0.14	0.03 0.97
P2P	352	422	0.48	352	421.84	0.36	352	422.1	0.48
	0.98 0.02	0.98 0.02	0.49 0.51	0.98 0.02	0.98 0.02	0.65 0.35	0.98 0.02	0.98 0.02	0.49 0.51
Streaming	2937	2,252	2.54	2,937	2,249.97	1.67	2,181	1846	1.64
	1 0	1 0	0.62 0.38	1 0	1 0	0.93 0.07	1 0	1 0	0.41 0.59
Voip	4,619	12,089	214.43	3,815	9,085	17.3	4,458	11,824	213.79
	0.18 0.82	0.08 0.92	0.01 0.99	0.21 0.79	0.11 0.89	0.13 0.87	0.15 0.85	0.06 0.94	0.01 0.99
Total	25,815	26,388	309.76	18,098	20,498	31.3	22,442	24,413	307.64
	0.82 0.18	0.52 0.48	0.02 0.97	0.83 0.17	0.6 0.4	0.23 0.77	0.79 0.21	0.48 0.52	0.02 0.98

TABLE II. SPECIFICATIONS OF THE DATASET WITH BOTH PREPROCESSING METHODS APPLIED

Category	By (Protocol, 3-handshake)		
	Total Packet Size (MB)	#Packet (K)	#Flow (K)
Chat	19	46.08	2.34
	0.71 0.29	0.66 0.34	0.13 0.87
Email	6	26.84	1.77
	0.56 0.44	0.71 0.29	0.22 0.78
File Transfer	8691	7524.59	7.31
	1 0	1 0	0.24 0.76
P2P	352	421.83	0.36
	0.98 0.02	0.98 0.02	0.65 0.35
Streaming	2181	1843.72	0.77
	1 0	1 0	0.85 0.15
Voip	3654	8819.62	16.63
	0.18 0.82	0.08 0.92	0.1 0.9
Total	14903	18682.69	29.2
	0.8 0.2	0.56 0.44	0.17 0.83

In this study, they reported extracting 11.6K bidirectional flows by removing 65% of flows that consisted of only one UDP packet and had a destination (IP address, port) equal to (255.255.255.255, 10505) from the overall flows. However, it should be noted that only flows generated from applications such as BlueStacks or Android emulators used for data collection were considered, and flows generated from the operating system or network configuration were not taken into account.

[3] used the ISCX VPN-nonVPN 2016 dataset as an initial study for deep learning-based traffic classification. However, they did not disclose the preprocessing steps involved and only provided information about the number of samples used. [4] proposes a method that transforms flow data into an intuitive picture called FlowPic and classifies it using a CNN-based model. In this study, they reported manually removing sessions that did not match the label category or removing incorrect packets, but the details were not clearly specified. In the current field of application traffic classification, there is a challenge in objectively comparing different studies due to the lack of clear criteria for data preprocessing and the limited availability of preprocessed datasets.

III. PREPROCESSING

This chapter describes the preprocessing methods of the dataset. The open dataset, ISCX VPN-nonVPN 2016, is composed of files (.pcap) for each application name and can be used for three tasks. First, we split the capture files into bidirectional flows (sessions) using the pcap splitting tool called *SplitCap* to extract information at the flow level. We perform data cleaning based on two criteria. The first criterion involves collecting the application layer protocols from each flow and determining whether these protocols are relevant to the behavior of the application. We remove flows that use protocols that may occur due to the operating system used or that are estimated to have occurred during network connections or communications. Table 3 shows the number of flows per protocol in each category, with red marking indicating the deleted protocols. It is suspected that the majority of the protocols were removed during the pre-processing stage for data collection or were used in the network configuration. Some of the removed protocols may have had close associations with specific applications. However, to ensure noise removal, any suspicious protocols were deleted.

Second, we remove flows using the TCP protocol in which the 3-way handshake is not preserved. If the offsets of packets within a flow are not consistent, it may disrupt the temporal sequence of packets within the flow when the model learns the traffic. Additionally, when applying the application classification model in a real environment, TCP traffic collected in real-time guarantees the preservation of the 3-way handshake, so there is no need to learn flows with disrupted 3-way handshake as noise. Therefore, we delete flows that have a packet count exceeding four (3-way handshake + data packets) and whose initial packet's tcp flag is not SYN, based on the aforementioned reasons.

The comparison of the dataset with the original dataset and the two preprocessing methods applied is shown in Table 1. Firstly, when preprocessing was performed based on protocols, the number of flows decreased from 309K to 31.3K. Additionally, examining the TCP/UDP ratio change revealed that a significant number of UDP flows were removed. Secondly, when preprocessing was performed based on the 3-way handshake, the number of flows decreased slightly from 309K to 307K. Lastly, when both preprocessing methods were applied, the number of flows reduced to 29.2K, and detailed specifications of the dataset are provided in Table 2.

IV. EXPERIMENT

In this chapter, we classify the original dataset and three preprocessed datasets using deep learning models, and

TABLE III. THE NUMBER OF FLOWS PER PROTOCOL WITHIN EACH CATEGORY

Networ layer	Application layer	CHAT	EMAIL	FILE_TRANSFER	P2P	STREAMING	VOIP	TOTAL
TCP	ancp			2				2
	bittorrent						3	3
	data	47	32	221	0	6	254	560
	dns					1	6	7
	ftp			22				22
	http	56	11	653	225	99	330	1,374
	nbss					10	17	27
	reload-framing			2				2
	rtmpt			1				1
	ssh			114				114
	stun						93	93
	tcp	57	19	641	2	593	332	1,644
	tls	361	401	463	8	854	1,283	3,370
	vnc		1					1
	x11	1		8				9
	xmpp			1				1
UDP	bjnp			5				5
	chargen						2	2
	data	2,010	1,353	5,563	127	18	14,651	23,722
	db-lsp-disc	42	42	42		5	65	196
	dcp-etsi			4			10	14
	dhcp	18	10	19			49	96
	dhcpv6	67	67	77			128	339
	dns	3,961	661	1,171	112	789	17,962	24,656
	dtls			1			14	15
	elasticsearch						2	2
	enip						1	1
	gquic	27	26	24		100	46	223
	kip						1	1
	llmnr	6,986	5154	60,795			177,837	250,772
	lsd	0	0	1			3	4
	mdns	66	58	88		4	172	388
	nbdgm	81	72	72		17	166	408
	nbns	122	119	193		32	453	919
	ntp	14	13	11	1	1	34	74
	nxp_802154_sniffer			1			1	2
	pathport						2	2
	portcontrol				1			1
	rtcp						2	2
	sip			2				2
	snmp	16	2	5			2	25
	srvloc	2	2	44			109	157
	ssdp	21	16	46	1	6	122	212
	stun						277	277
	teredo			5		0	5	10
Total		13,955	8,059	70,297	477	2,535	214,434	309,757

compare and analyze the experimental results. [5] is a deep learning model that utilizes an ensemble technique extracting various features from the shapes derived from the input traffic and is used for comparison. The experimental dataset is divided into a training set and a test set in an 8:2 ratio.

Table 4 represents the accuracy when applying preprocessing methods sequentially to the original dataset. The accuracy is divided into OA(overall accuracy) and BA(balanced accuracy), and when preprocessing with the protocol and 3-handshake, it results in the highest accuracy.

TABLE IV. ACCURACY BY PREPROCESSING

	#Flow	OA	BA
Original	309.76K	85.2%	76.9%
Protocol	31.3K	93.5%	89.5%
3-hand-shake	307.64K	87.8%	82.6%
Protocol+ 3-handshake	29.2K	95.7%	93.8%

Table 5 shows the results when creating an inference model for each dataset using the training dataset and applying it to the test dataset where the 3-handshake is preserved. As mentioned before, we assume that the data

TABLE V. ACCURACY BY PREPROCESSING

	Training	Test	OA	WA
Original	247.8K	5.84K	85.2%	71.5%
Protocol	25K		93.5%	91.4%
3-hand-shake	246.1K		87.8%	72%
Protocol+ 3-handshake	23.4K		95.7%	93.8%

collected in real environments guarantees the preservation of the 3-handshake in TCP flows. The experimental results indicate that the model trained on the training dataset with preserved 3-handshake achieves the highest performance. This implies that using data with preserved 3-handshake when preprocessing application traffic analysis datasets is appropriate.

V. CONCLUSION

This paper conducts preprocessing and analysis on the widely used ISCX VPN-nonVPN 2016 public dataset in the field of application traffic classification. Additionally, it makes the preprocessed dataset publicly available to enable objective performance comparisons with other papers that utilize this public dataset. The preprocessed dataset, dataset structure, detailed preprocessing steps, and accompanying code can be accessed through the provided link (https://github.com/pb1069/preprocessing-of-ISCX-VPN-nonVPN-2016).

REFERENCES

[1] DRAPER-GIL, Gerard, et al. Characterization of encrypted and vpn traffic using time-related. In: Proceedings of the 2nd international conference on information systems security and privacy (ICISSP). 2016. p. 407-414.

[2] ACETO, Giuseppe, et al. DISTILLER: Encrypted traffic classification via multimodal multitask deep learning. Journal of Network and Computer Applications, 2021, 183: 102985.

[3] WANG, Wei, et al. End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In: 2017 IEEE international conference on intelligence and security informatics (ISI). IEEE, 2017. p. 43-48.

[4] SHAPIRA, Tal; SHAVITT, Yuval. FlowPic: A generic representation for encrypted traffic classification and applications identification. IEEE Transactions on Network and Service Management, 2021, 18.2: 1218-1232.

[5] BAEK, Ui–Jun, et al. MISCNN: A Novel Learning Scheme for CNN-Based Network Traffic Classification. In: 2022 23rd Asia-Pacific Network Operations and Management Symposium (APNOMS). IEEE, 2022. p. 01-06.