

A novel approach for application classification with encrypted traffic using BERT and packet headers

Jaehak Yu, Yangseo Choi, Kijong Koo, Daesung Moon*

Intelligent Network Security Research Section, Cyber Security Research Division, Electronics and Telecommunications Research Institute (ETRI), 218 Gajeong-ro, Yuseong-gu, Daejeon 34129, South Korea

ARTICLE INFO

Keywords:

Encrypted traffic classification
Network security monitoring
Bidirectional encoder representations from transformers
BERT
Internet application classification

ABSTRACT

Recent years have seen substantial advancements in Internet technology along with environmental changes, which have led to the emergence of various security issues. There is also a trend of explosive growth in applications that encrypt network traffic for various types of services. Therefore, the classification of applications within encrypted traffic represents an important research issue for both secure network management and efficient bandwidth management. In such encrypted traffic, the payload itself is encrypted, and it is no longer viable to classify applications based on signatures extracted from plaintext. Most applications in public datasets for encrypted traffic classification are collected with the same IP address and port number, which makes the 5-tuple information a strong identifier. However, this 5-tuple contains many characteristics related to both the traffic collection environment and user-specific traits, rather than intrinsic features of the applications themselves. Therefore, when addressing the problem of encrypted traffic application classification, it is advisable to utilize header information excluding the 5-tuple and payload. Therefore, this paper proposes a novel service type and application classification system based on the Bidirectional Encoding Representation Transformer (BERT), which utilizes packet header information from encrypted traffic. The proposed system ensures the accuracy and generalization performance of the classification model by using only the header information from traffic packets, excluding the 5-tuple and payload. Further, to preserve the characteristics and semantic meaning of an encrypted traffic packet, sentences embedded with 2-byte tokens were used as input for BERT. The proposed system was designed to exclude labeling information from all sentences during the pre-training phase before proceeding with training. Fine-tuning was then conducted to align the system with the objectives of the service type and application classification. This experiment utilized the publicly available ISCX VPN-nonVPN dataset, and the proposed model achieved remarkable accuracy in the key performance measure, i.e., F1-scores, with values of 99.24 % in service type classification and 98.74 % in application classification. This capability can be used in maintaining the confidentiality of encrypted traffic, network security monitoring, Quality of Service (QoS), and traffic management in complex IT environments.

1. Introduction

The continual advancement of the Internet and its corresponding increase in users has led to the provision of a wide array of services and content through the Internet [1-3]. Network traffic classification involves identifying the categories of traffic originating from various web services or applications, therefore making it one of the crucial research areas for ensuring a seamless network environment and enhancing network security [4-6]. This particular field of traffic classification plays an important role in stable network management, proper support of

Quality of Service (QoS), and efficient bandwidth management, among other contexts [7-8]. Traffic encryption technology has recently become to be widely used to protect the personal information of Internet users to ensure confidentiality. In particular, with growing interest from both individuals and businesses, the application of encryption protocols such as Transport Layer Security (TLS) and Secure Sockets Layer (SSL) to web services and Internet application data is becoming increasingly prevalent [8-9]. While the use of encryption protocols provides advantages such as protecting personal information and ensuring anonymity for Internet users, such protocols are increasingly coming to pose a

* Corresponding author.

E-mail addresses: dbzzang@etri.re.kr (J. Yu), yschoi92@etri.re.kr (Y. Choi), kjkoo@etri.re.kr (K. Koo), daesung@etri.re.kr (D. Moon).

<https://doi.org/10.1016/j.comnet.2024.110747>

Received 25 March 2024; Received in revised form 30 July 2024; Accepted 22 August 2024

Available online 1 September 2024

1389-1286/© 2024 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

challenge to network and resource management stemming from the classification of encrypted traffic [10-14]. Therefore, accurately classifying diverse encrypted traffic applications is one of the essential research challenges for stable network and system management [15].

Recent surveys of research in this area have shown that the use of applications based on encrypted traffic for bolstering the protection of personal information and securing anonymity is on the rise [16-18]. In early studies, a fingerprint was generated from plaintext before the traffic was encrypted, after which the traffic was classified by matching a pre-defined pattern [19-23]. Deep Packet Inspection (DPI) [16,24] and FlowPrint [19] notably employ this fingerprint method, where DPI utilizes the entirety of the packet, including the header and the payload. However, this method only involves using plaintext information, which shows limited classification accuracy, as it is susceptible to tampering or loss during transfer. In subsequent studies, statistical feature values were extracted and picked from traffic flow, after which classical machine learning algorithms such as the decision tree, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) were employed during the traffic classification methods [18,23]. Various attempts have recently been made to classify the raw data of encrypted traffic using deep learning models [10-11,24]. Such research has typically been conducted on an expanded or improved encrypted traffic classification model based on Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). However, these previous studies have been highly dependent on the amount and distribution of labeled training data, which can lead to biases in classification models. Moreover, the RNN model which is suitable for processing time-series data suffers from declining training performance as the input sentence increases in length. To mitigate these shortcomings of the CNN and the RNN, some ongoing research is focused on Transformers [9], which learn the sequential relationships and dependencies between words in a sentence [10-11]. Instead of using an encoder-decoder model with self-attention in the Transformer, there has also been active research in encrypted traffic classification using Bidirectional Encoding Representations Transformers (BERT) pre-trained on large-scale language models [2,9-11,25]. In this approach, the BERT model is pre-trained using large-scale unlabeled network traffic data, then fine-tuned using labeled data to be used in encrypted traffic classification. Such BERT models construct sentences from traffic packet header fields and are trained with the aim of accurately capturing the bidirectional relationships and contextual meanings of these header fields. BERT also has a structure where it is pre-trained on large-scale text data and then fine-tuned for specific tasks. This allows for incremental updates to be achieved without needing to pre-train the entire model again, specifically by fine-tuning the model to include newly added encrypted applications. There has recently been growing interest in using BERT's advantages for encrypted traffic application classification research. For example, Lin et al. [10] and Shi et al. [11] reported that the BERT model can achieve better performance than existing encrypted traffic classification methodologies. However, the BERT models proposed by Lin et al. [10] and Shi et al. [11] use the 5-tuple of packet headers and randomly encrypted payloads, which makes it difficult to guarantee performance when the network topology changes or new applications are added. Since the payload of encrypted traffic is randomly encrypted, even the same application can have different payloads, making it challenging to use as a feature for classification models. Moreover, in real-world environments, applications dynamically allocate Internet Protocol (IP) address, while the port number assigned to clients connecting to servers is randomly assigned by the operating system (OS); the port number changes with each connection. Therefore, using 5-tuple information, such as IP address and port number from public datasets, as features results in models that are specialized to those datasets, with significantly reduced generalization performance and accuracy. It is thus advisable to use header information excluding the 5-tuple and payload for the problem of encrypted traffic application classification.

In the current paper, we propose a novel service type and application

classification system that is based on the BERT using the packet header information of encrypted traffic. The paper makes several key contributions:

- 1) The proposed model ensures that encrypted traffic is accurately classified by extracting 2-byte tokens from the header fields, excluding the 5-tuple and payload, and creating sentences that preserve the unique characteristics and context of the packet header fields. We performed pre-training and fine-tuning on these generated sentences with the aims of maintaining the properties of the language understanding model BERT and enhancing classification performance.
- 2) By using only the header field information, excluding the 5-tuple and payload, we ensured the generalization performance of the encrypted traffic application classification model and contributed to protecting user privacy. Recent applications use dynamic IP allocation and port number alternation techniques, making the use of the 5-tuple a major cause of bias and reduced classification performance in encrypted traffic classification models. Moreover, since encrypted traffic has randomly encrypted payloads, it is difficult to ensure the learning and generalization performance of the classification model is challenging. Therefore, by removing the 5-tuple and payload, the proposed system can be applied to real-world environments while being adaptable to changes in network environments.
- 3) The proposed system was designed to facilitate fine-tuning tailored to specific classification objectives by service type and application, and the ISCX VPN-nonVPN dataset was used for the classification experiment. Our system allows for incremental updates: Even when new applications are added, our system only requires fine-tuning without the need for pre-training. The key performance metric F1-score was achieved with a high classification accuracy of 99.24 % in service type classification and 98.74 % in application classification.
- 4) The BERT-based encrypted traffic classification model proposed in this paper can be used in a number of ways, with possible applications including the protection of confidentiality for encrypted traffic, network security monitoring, anomaly detection, traffic management in complex IT environments, and implementing proper Quality of Service (QoS).

The rest of this paper is structured as follows: [Section 2](#) will examine related works on the application classification of encrypted traffic, while [Section 3](#) will describe the service type and application classification system based on the packet header information proposed in this paper. [Section 4](#) will present experiment results and performance analysis, and finally, [Section 5](#) will present the conclusion and discuss future research directions.

2. Related work

2.1. Encryption traffic classification studies

Previous studies on network encrypted traffic classification can be separated into those using port number-based methods, fingerprint construction methods, statistical methods, deep learning models, and pre-training models.

- 1) **Port number methods:** The port number-based encrypted traffic classification methods are among the most classical methods. These methods involve assigning specific port numbers to each application and classifying traffic based on that information [26]. As an example, Secure Shell (SSH) protocols are assigned to 22, while Hypertext Transfer Protocol (HTTP) protocols to 80. Port number-based application classification methods, like those employed in Access Control Lists (ACLs) or firewalls, utilize port information from the transport layer to efficiently classify network traffic, ensuring both

cost-effective and highly accurate operation. Additionally, Peer-to-Peer (P2P) networks and some streaming services do not use a fixed port number but instead use a variety of port numbers, which makes it difficult to accurately classify traffic based on port number alone. Due to the recent growth in port number alternation, technology and applications that randomly set port numbers, the classification accuracy inevitably decreases when applied to real-world network environments. Therefore, this paper attempted to classify encrypted traffic without relying on 5-tuple information, which includes the port number.

- 2) **Fingerprint construction methods:** Fingerprint construction methods generate a fingerprint from a plaintext or Transport Layer Security (TLS) handshake before the network traffic is encrypted, and the traffic types are then classified by matching the traffic against pre-defined string patterns [19-21]. Widely known methods for this purpose include FlowPrint [19-20] and Deep Packet Inspection (DPI) [24]. FlowPrint classifies traffic types by utilizing unencrypted protocol field information such as size, certificate, time statistical values, and more from packets originating within the flow. Meanwhile, DPI utilizes the entirety of the packet, including the header information and the payload, and it performs classification by detecting the pre-defined signature. These two fingerprint construction methods excel in accurately classifying the traffic type within the initial packets of a flow. However, since they use plaintext information, the traffic is vulnerable to tampering during transfer as well as prone to data loss. Therefore, it is difficult to extract pre-defined string patterns using fingerprint construction for encrypted traffic and use them for application classification. Moreover, the usage of plaintext has recently been declining in network environments, and encryption technologies such as Transport Layer Security (TLS) 1.3 complicate traffic classification methods that use fingerprints.
- 3) **Statistical methods:** Statistical methods define and extract statistical properties from encrypted traffic and classify traffic through classical Machine Learning (ML) algorithms [18,22,27-28]. AppScanner [27] used the statistical characteristics of packet size and attempted to classify encrypted traffic by training with the Random Forest (RF) model. Meanwhile, BI-directional Dependence (BIND) [28] uses dependent statistical values of the interval length in a packet to classify traffic. ML algorithms based on statistical properties offer the advantage of lower computational complexity, as they are not required to inspect every bytes of packet content, nor do they require access to the packet's content to classify the encrypted traffic; instead, they rely solely on data or statistical property values. However, when using statistical information, it is highly possible that the extracted signatures will be dependent on the protocol or engine used by specific applications, thus making them unsuitable for encrypted traffic classification. The involvement of the subjective judgment of experts in defining features, selecting statistical value extraction algorithms, and determining labeling can also lead to significant variations in classification accuracy.
- 4) **Deep learning models:** Recent deep learning models automatically extract and train with complex patterns and raw features from network traffic, and they yield highly accurate encrypted traffic classification and identification results [10-12,29-32]. For example, Sirinam et al. [33] used the Convolutional Neural Network (CNN), while Liu et al. [34] used the Recurrent Neural Network (RNN) to automatically extract traffic features from a raw packet-sized sequence to classify traffic. Lotfollahi et al. [29] and Lin et al. [34] combined feature extraction from deep learning models and classifiers into a single architecture. Next, features were extracted from the network packet header and payload before encryption, and the network traffic was classified. However, deep learning models not only require large-scale labeled training data, but they can also present biased classification results if trained with unbalanced datasets. In this paper, we were able to achieve pre-training without

the need for large-scale labeled training data or appropriate fine-tuning to a desired classification purpose, which ultimately resulted in high classification performance.

2.2. BERT-based pre-training models

Initially, with advancements in deep learning technology, studies in the field of computer vision using CNN [33] and in natural language processing using RNN [34] have demonstrated high performance, with over 90 % accuracy in encrypted traffic classification. Subsequently, the Transformer [9-11] model, which is more advanced than RNN in the field of natural language processing with time-series continuity, was introduced, and it has since come to be actively used in various fields. These Transformer models have proven useful not only in natural language processing but also in the field of computer vision. Consequently, research utilizing these models has also been initialized in the field of network encrypted traffic classification, which consists of time-series continuous packets [2,10-11]. BERT has demonstrated high performance across various fields, including language understanding, question answering, and other natural language processing tasks, thus causing it to be widely adopted in the research and industry domains. Pre-trained BERT models have been shown to have the ability to extract generalized features from large-scale unlabeled training data depending on the desired purpose, while they have also been shown to achieve excellent performance when fine-tuned with small-scale labeled training data [35-36].

BERT repeats a process of prediction by randomly masking parts of tokens via an unsupervised Masked Language Model (MLM) during pre-training, after which it fine-tunes the learning model, thus enabling its application to multiclass problems depending on the problem definition [2,10-11]. Although the payload of a packet is not explicitly meaningful in encrypted traffic, Sengupta et al. [37] conducted a study on traffic classification and identification using the difference in randomness between various encrypted traffic. In another study, He et al. [38] proposed Payload Encoding Representation from Transformer (PERT), which represents the first packet payload of a traffic flow as pixel values in a grayscale image, tokenizes it into 2-byte words (tokens), and generates sentences for training. PERT achieved an F1-score of 93.23 % on the ISCX VPN-Service dataset, thus outperforming the method proposed by Lan et al. [39]. However, PERT proposed by He et al. [38] has limitations in explaining the characteristics of encrypted traffic and the pre-training process, which ultimately results in reduced generalizability of the classification model. Hu et al. [40] proposed a model composed of a 1D-CNN and a Transformer encoder. They attempted to classify encrypted traffic by pre-training with unlabeled datasets and transferring the learning to another dataset.

Recently, Lin et al. [10] proposed ET-BERT, which uses the Datagram2Token module to extract packets from a single session flow and generates sentences for the BERT model by extracting 2-byte tokens. Experimental results obtained on five public datasets, including ISCX VPN-nonVPN, showed desirable performance metrics with F1-scores of 98.90 % and 99.37 %, respectively. ET-BERT [10] only uses 86 payloads in 2-byte units, excluding the 5-tuple and headers. However, as highlighted in the study by Sengupta et al. [37], there are limitations associated with using payloads, since the randomness is not perfect and implicit identification is possible. Shi et al. [11] proposed BFCN, which uses sentences generated with the same tokenization as ET-BERT [10]. It combines CNN layers with BERT to simultaneously consider byte-level local features. However, like ET-BERT, it removes all packet header information, including the 5-tuple and it uses only randomly encrypted payloads, thus showing no significant difference from ET-BERT in this regard. These prior studies have failed to overcome the limitation of relying solely on payloads, which are not perfectly random and can be implicitly identified, while utilizing the inherent characteristics and meaningful features of encrypted traffic packets instead maybe be a desirable alternative.

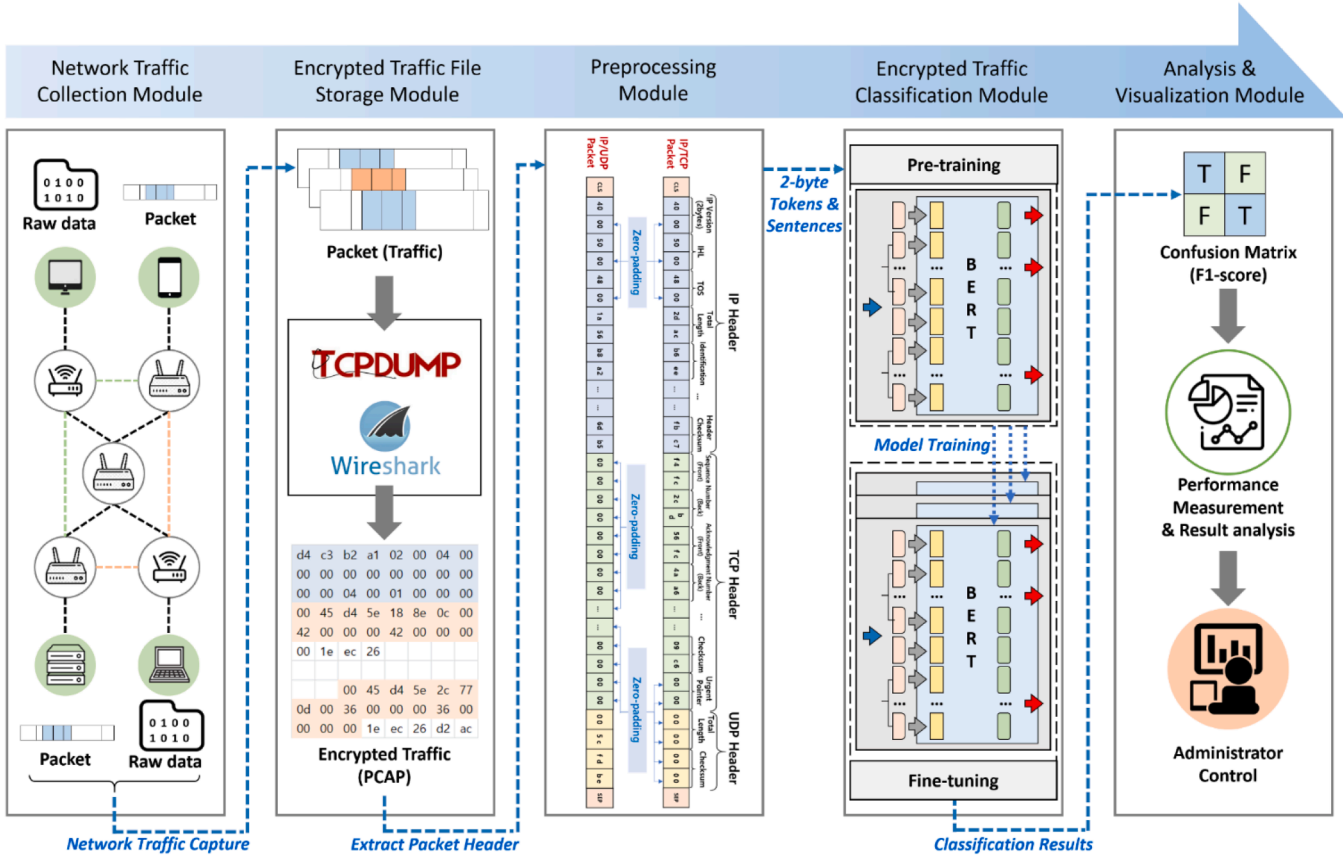


Fig. 1. BERT-based encrypted traffic classification system.

3. Proposed system structure

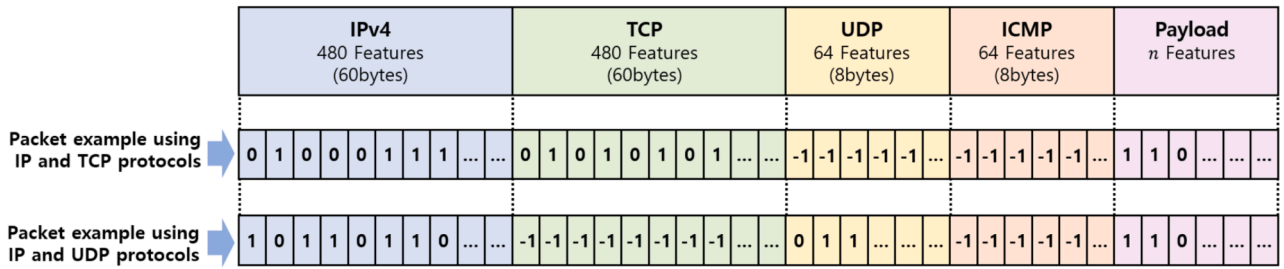
The current paper provides a detailed explanation of a BERT-based application classification system that utilizes the context and semantic information of packet header fields in encrypted traffic. The proposed system removes the 5-tuple of packet headers since identification information such as IP address or port number can lead to the introduction of biases in training models. The proposed BERT model tokenizes packet header field values into 2-byte units for each application to generate sentences for pre-training and fine-tuning. In this process, the 5-tuple and payload of the packet header are not used, in an attempt to prevent model overfitting and performance degradation. First, if the packet header fields are divided into 1-byte units, their unique characteristics disappear. Conversely, if they are represented in 4-byte units, independent adjacent field values become combined, which causes a loss of semantic information. Moreover, representing tokens in 4-byte units exponentially increases the vocabulary size of BERT, which significantly extends the time required for pre-training and fine-tuning. Therefore, in this study, packet header fields are expressed in 2-byte units, as has been done in previous methods [10-11]. However, unlike [10], to preserve the unique characteristics and information of network traffic packets, the sequence number and acknowledgment number are tokenized in a fixed order. The structure and functions of the proposed system, from network traffic collection to pre-training, fine-tuning, classification results, and analysis modules, are detailed in the following (refer to Fig. 1 below).

1) **Network Traffic Collection Module:** This module collects packets using tools like Tcpcdump [Anon., 41] and Wireshark [Anon., 42], which are applications for monitoring and capturing network traffic. For instance, Tcpcdump can be configured to dump packets based on specified options such as host, net, and port types. It can also capture

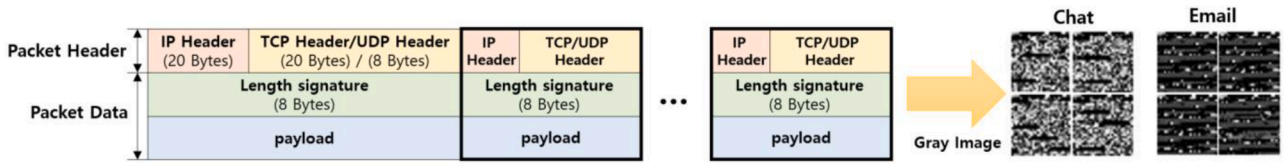
network traffic by setting the IP header's source and destination in a unidirectional or bidirectional manner. Packets can also be collected by selecting protocols such as ethernet, IP, Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Internet Control Message Protocol (ICMP). The collected network traffic packets are then categorized by service type (e.g., Email, Streaming, VoIP) and into individual applications before being transferred to the encrypted traffic collection file storage module.

2) **Encrypted Traffic File Storage Module:** Packets captured and received from the network traffic collection module are decoded from the data link layer to the application layer and stored as files. This module saves these packets in Packet Capture (PCAP) format or Packet Capture Next Generation (PCAPNG) format. For example, the PCAP format includes a header at the beginning of the file that defines the format, and it includes information such as version, timestamp, and network type. Each packet header is followed by the actual packet data. While the PCAP format stores packets in a simple structure, PCAPNG allows for various types of information to be stored in a block format due to its extensible structure. These stored PCAP or PCAPNG files are then transferred to the next preprocessing module as appropriate.

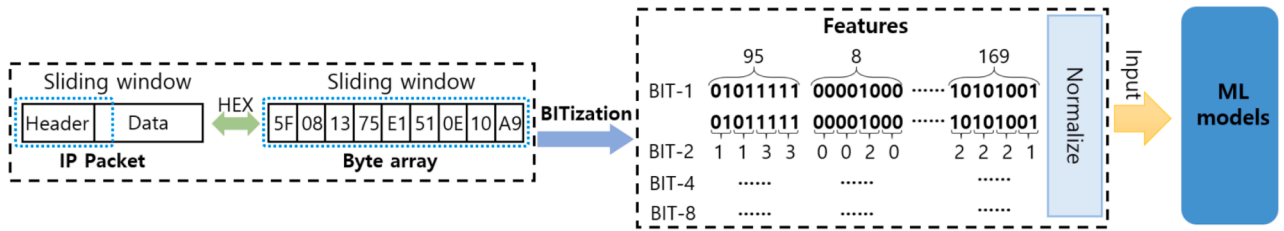
3) **Preprocessing Module:** In this module, packet headers are first extracted from the PCAP or PCAPNG files, after which the 5-tuple and payload are removed. The values of each header field are then converted into 2-byte hexadecimal tokens, which are used to create sentences for BERT's pre-training and fine-tuning. The sequence number and acknowledgment number fields of the TCP header are 4-byte, meaning every header's field values can be represented in 4-byte. However, if every header field is represented as 4-byte, the vocabulary size in the BERT model will increase exponentially and require substantial time costs in pre-training and fine-tuning; furthermore, each header field will have an increased amount of



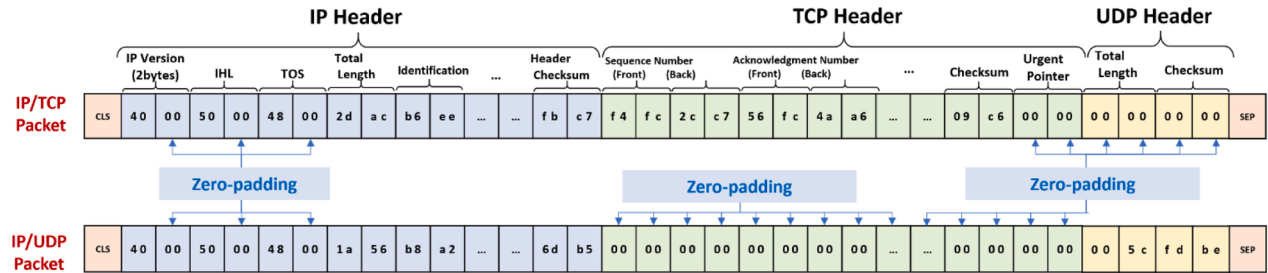
(A) A single-packet representation (nPrint) [20]



(B) Packet structure of the traffic reconstruction [22]



(C) BITization features (BIT-1, BIT-2, BIT-4, and BIT-8) [43]



(D) Our proposed method

Fig. 2. Comparison of our proposed method using packet header information with approaches described in previous studies.

zero padding, ultimately leading to a high probability of loss of context and semantic information. Consequently, this module represents the values of packet header fields as 2-byte tokens and sequentially combines them to form sentences. During this process, the 4-byte sequence number and acknowledgment number in the TCP header are divided into 2-byte units while ensuring that their original order is maintained.

4) **Encrypted Traffic Classification Module:** In this module, pre-training and fine-tuning of BERT are performed using sentences that have been created from packet headers by the preprocessing module. During the pre-training phase, the model is trained to predict the next word or token based on the previous words or tokens. In this phase, labels are excluded from all of the sentences that are used for pre-training, and a certain percentage of input tokens is randomly masked. In this study, 15 % of the tokens in a sentence were

randomly masked. If the masking ratio is too high, it becomes difficult for the model to learn the context; if it is too low, it takes a long time to converge. After pre-training, fine-tuning is conducted using a small amount of labeled data that had not been used in pre-training and which was tailored to the classification objectives by service type or application. This fine-tuning process optimizes the model for the specific purpose of classifying encrypted traffic by service type or application, which ultimately results in improved classification performance.

5) **Analysis and Visualization Module:** This module aims to classify encrypted traffic by service type and application, thus providing the fine-tuning results and classification performance metrics such as accuracy, F1-score, and confusion matrix. By offering a comprehensive analysis and visualization of information about encrypted traffic, this module helps network administrators understand

misclassifications as well as their causes. This serves as scientific analysis that can be applied to facilitate network security monitoring, Quality of Service (QoS), and traffic management in complex IT environments.

Previous studies extracted packet header fields not as semantic values but as successive feature values, or they generated images as input data (refer to (A), (B), and (C) in Fig. 2). Holland et al. [20] proposed a method where the 5-tuple values are removed, and where the header fields and payload are represented as binary data (0 s and 1 s) for use as inputs to machine learning models. The authors of that study assumed that the packet itself has a specific semantic structure, and they suggested a method that could be used to arrange header information and user-defined n payloads as binary data. First, as shown in Fig. 2(A), they presented an example of a single packet using IP and TCP protocols. The value for the IP protocol was set to 01000111, and the value for the TCP protocol was set to 01010101. All unused values for UDP and ICMP were set to -1. If the value of n for the payload was 10, the binary representation was generated as 1101100010. The generated binary values were then used for training to classify application types using machine learning or Auto Machine Learning (AutoML) techniques. Next, as shown in Fig. 2(B), Ma et al. [22] proposed a method that could be used to generate a grayscale image by extracting the IP, TCP, and UDP header information of packets, an 8-byte length signature, and bytes from the 25th to the 100th bytes of the payload. They then created a CNN-based learning model using these images to perform encrypted traffic classification. Ma et al. [22] noted that 5-tuple values can lead to overfitting, and they set them all to 0. Luo et al. [43] proposed a feature transformation method called BITization (including BIT-1, BIT-2, BIT-4, and BIT-8) to generate input features for machine learning (see Fig. 2 (C)). BITization first converts the packet header and data into byte level hexadecimal values. These values are then extracted into four different feature sets: BIT-1, BIT-2, BIT-4, and BIT-8. For example, the hexadecimal value 5F is converted to 01011111 in BIT-1. In BIT-2, the bits from BIT-1 are paired to produce the values 1, 1, 3, 3. In BIT-2, the bits from BIT-1 are grouped into sets of four to produce the values 5 and 15. In BIT-4, the bits from BIT-1 are grouped into sets of eight to produce the value 95. Lastly, the extracted values for each BIT are normalized and used as input data for machine learning to classify encrypted traffic.

In this paper, as presented in Fig. 1, the process from network traffic collection to PCAP file storage, extraction of header fields in 2-byte units, pre-training and fine-tuning for encrypted traffic classification, and analysis and visualization of classification results can provide network monitoring and management information. In particular, an example of a packet header field being embedded in the form of 2-byte tokens to maintain the semantic definition as proposed by this paper can be found in (D) of Fig. 2, and unused fields had zero padding and were used as input sentences for BERT. However, the sequence number and acknowledgment number in the TCP protocol have maximum lengths of 4 bytes, and they are divided into 2-byte lengths with fixed sequences to maintain the context in the BERT. In the BERT model proposed in this paper, the resulting token embedding is merged with position embedding and segment embedding, after which layer normalization and dropout are applied. After this process, it is then used as an input for the bidirectional transformer. While BERT typically utilizes two learning methods for pre-training, this paper exclusively utilizes the Masked Language Model (MLM) approach. MLM applies a 15 % probability of transforming tokens from the input sequence into mask tokens, along with a prediction of mask tokens before they are transformed by the language model. The experiment in this paper separates the purpose of classification by service type in Section 4.3 and each application classification task in Section 4.4, and then fine-tunes appropriately.

Table 1
Dataset used in our experiment.

Dataset	Services	Applications
ISCX VPN-nonVPN	Chat	AIM-Chat
	Email	Email
	File-Transfer (FT)	Facebook
	Streaming	Gmail
	VoIP	Hangout
	VPN—Chat	ICQ-Chat
	VPN-Email	Netflix
	VPN-FT	SCP
	VPN-P2P	Skype
	VPN-Streaming	Spotify
	VPN-VoIP	Vimeo
		VoipBuster
		VPN-Ftps
	VPN-Sftp	
	Youtube	

Table 2
PCAP files included by service types.

Dataset	Services Types	PCAP file list
ISCX VPN	VPN—Chat	aim_chat, facebook_chat, skype_chat, hangouts_chat,
	VPN-Email	icq_chat
	VPN-FT	email, gmail
	VPN-P2P	skype_files, ftps, sftp
	VPN- Streaming	bittorrent
	VPN-VoIP	vimeo, youtube, netflix, spotify facebook_audio, hangouts_audio, skype_audio, voipbuster

4. Experimental results

4.1. Datasets and data preprocessing

In this Section, we measure the performance of the BERT-based encrypted classification system using the public ISCX VPN-nonVPN traffic dataset that is made publicly available by the University of New Brunswick [44-Anon., 45]. The VPN-nonVPN dataset consists of representations of real-world traffic generated in ISCX, including raw traffic data generated from various application programs, both VPN and nonVPN encrypted. This dataset provides labeled network traffic in the form of PCAP files, which are categorized by the service type (such as Chats, Emails, File Transfers, Streaming, etc.) provided by the application when the traffic was captured as well as the application that generated the traffic (e.g., Gmail, Facebook, Skype, Voipbuster, Youtube, etc.) [44-Anon., 45]. In this experiment, we classify the ISCX VPN-nonVPN dataset into 11 service types and 15 applications, which are detailed in Table 1. For the comparative experiment of the methodology proposed in this paper, the previous research [10-11] on encrypted traffic classification was used as a reference. In the experiment of this paper, Peer-to-Peer (P2P) from the service types, Tor from applications, and Torrent from nonVPN, which were not included in the ISXC VPN-nonVPN dataset, were excluded [10].

The ISCX VPN-nonVPN dataset totals 28 GB and consists of 34 and 109 PCAP files of VPN and nonVPN traffic, respectively. In this paper, packet header information was extracted from the PCAP files of each application; Table 2 presents a list of six PCAP files by service type in the ISCX VPN traffic [44-Anon., 45].

In this paper, 5-tuple information, including the IP address, protocol, and port number of TCP and UDP, were removed from the packet header field. Depending on the network environment, 5-tuple information may appear either identical or similar. As a result, IP and port number can serve as strong identification information that leads to a high probability of biases in the learning model and overfitting of the classification model. Therefore, for the BERT-based service type and application classification in this paper, 5-tuple information was removed. In the UDP

Table 3
Example header field values for 11 service types.

IP version	IP header length	IP TOS	IP total length	IP flags	...	UDP length	UDP checksum	Label name
4000	5000	4800	0550	c8d7	...	0077	1326	VPN-Streaming
4000	5000	0000	0028	77ad	...	004d	66f2	nonVPN-FT
4000	5000	0800	0034	123e	...	0035	a9a1	VPN-P2P
4000	5000	0000	0034	7c53	...	0000	0000	VPN-Email
4000	5000	0000	0065	d4fa	...	0051	e89e	nonVPN-Voip
4000	5000	0000	0034	9040	...	0000	0000	VPN-Chat
...
4000	5000	4800	0058	0000	...	0044	e451	nonVPN-Chat
4000	5000	0800	0034	ea33	...	0000	0000	nonVPN-FT

Table 4
Statistical information of dataset extracted from ISCX VPN-nonVPN.

Dataset	Classification tasks	Number of total samples	Number of labels
VPN-	Service types	1100,000	11 types
nonVPN	Each application	1297,134	15 types

protocol, source, destination port, and payload were all removed, and only total length and checksum header fields were extracted for use. Table 3 presents an example of the packet header fields utilized in the classification experiment into 11 service types outlined in Table 1 extracted in the form of 2-byte values with 5-tuple excluded.

In this paper, only the packet header information of each protocol was extracted and used for the classification by service type and applications based on the BERT model. Each set of header information was used in the form of independent 2 byte input tokens, and any fields that were configured by the system to be smaller than 2-bytes (e.g., IP version, IP header length, IP TOS, TCP window, total length of UDP, etc.) were zero padding. Processing the packet header field value as a fixed-length bit or byte rather than semantically would lead to values of different fields merging or single fields dividing into several tokens, thus resulting in lost unique features. We observed this in detail when reviewing the previous research in Fig. 2 of Section 3 [20-22]. In the experiment where a 4-byte input token was used for the BERT model proposed in this paper, >580 million vocabularies were generated, and restrictions such as increased learning time during pre-training and fine-tuning, excessive memory usage, etc. were noted. Therefore, in this paper, 2-byte tokens were used as vocabularies to maintain the semantically unique feature of the traffic packet header. The execution of the pre-training and fine-tuning of the BERT then used the sentences embedded by those tokens.

For the classification experiment that involved classification into 11 service types, a maximum of 100,000 packets was randomly extracted from each class. The entire dataset was divided into training, validation, and test datasets in an 8:1:1 ratio, respectively. Similarly, for the classification experiment into 15 application types, a maximum of 100,000 packets was also randomly extracted. Classes with <100,000 samples, such as spotify, email, ICQ_chat, gmail, and AIM-chat, were likewise used in their entirety. Table 4 presents the statistical information of the dataset per service and application extracted from the ISCX VPN-nonVPN dataset.

4.2. Experiment configuration and evaluation metrics

The experimental environment used in this paper consists of an OS running Ubuntu 20.04.4 LTS, an AMD Ryzen Threadripper Pro 5975WX CPU with 64 cores, a single NVIDIA Quadro RTX A6000 GPU, and 256 MB of RAM. To validate the classification results of service and application types based upon the ISCX VPN-nonVPN dataset and to assess the system performance, four performance evaluation metrics were used; details are shown in formulas (1) to (4) [10-12,46-47]. In this instance, accuracy measures the proportion of correct predictions among the samples processed by the classification sample. While accuracy is the

simplest metric for evaluating a classification model, it is difficult to make objective performance evaluations in this experimental environment due to the presence of unbalanced classes in the service and application datasets. Consequently, the current paper used the additional performance evaluation metrics of F1-score, recall, and precision. The F1-score is a harmonic mean of recall and precision, and it has a value between 0 and 1, where a value closer to 1 indicates higher classification performance [46-47]. Meanwhile, recall is the proportion of samples declared positive by the classification model among the actual positive samples, and it indicates the ability to retrieve the actual positive classes without loss. Lastly, precision is the proportion of the actual positive samples among the samples declared positive. When evaluating a classification model, a standard approach involves processing target types or applications as positive classes and other types or applications as negative classes. While it is crucial to achieve high accuracy in a classification model, to optimize the performance of service and application classifiers in encrypted traffic, it is necessary to train the model to minimize false positives and false negatives.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \quad (1)$$

$$F1 - score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

4.3. Classification into 11 service types

The first experiment executed the classification of six types of VPN and five types of nonVPN, thus classifying a total of 11 types according to the service of encrypted traffic. The datasets are constructed in an identical manner to that used in precedent papers [10-11,48-49] for the comparative experiment, with nonVPN P2P services excluded from the experiment due to the lack of PCAP files provided in the ISCX VPN-nonVPN. We experimented with sequence lengths ranging from 8 to 512 (8, 16, 32, 64, 128, 256, 512) as well as batch sizes ranging from 8 to 512. Shorter sequence lengths in BERT reduce resource consumption and increase training and inference speed, but they may lose context. Conversely, longer sequence lengths capture better context and improve classification performance, but they require more computation time and memory. Our experiments showed that a sequence length of 64 and a batch size of 32 yielded the most stable classification model performance. The total pre-training steps were tested within a range from 100,000 to 2000,000, and we ultimately settled on a fixed number of 1000,000. Fine-tuning epochs were precisely adjusted from 5 to 20, and after repeated trials, 10 was chosen as the default. Lastly, the learning rate was set to 2×10^{-5} , and AdamW [50] was chosen as the optimizer so that the experiment could be performed under identical parameters.

In this paper, for the purpose of conducting an experiment that could

Table 5
Comparison results on classification of service types.

Method	Accuracy	Precision	Recall	F1-score
AppScanner [27]	71.82	73.39	72.25	71.97
CUMUL [23]	56.10	58.83	56.76	56.68
BIND [28]	75.34	75.83	74.88	74.20
FlowPrint [19-20]	79.62	80.42	78.12	78.20
FS-Net [31]	72.05	75.02	72.38	71.31
GraphDApp [32]	59.77	60.45	62.20	60.36
DeepPacket [29]	93.29	93.77	93.06	93.21
PERT [38]	93.52	94.00	93.49	93.68
BFCN [11]	99.12	99.13	99.11	99.11
ET-BERT (flow) [10]	97.29	97.56	97.31	97.33
ET-BERT (packet) [10]	98.90	98.91	98.90	98.90
Proposed	99.25	99.26	99.24	99.24

Table 6
Performance evaluations of service types classification.

Service category	BFCN [11]			Proposed		
	precision	recall	F1-score	precision	recall	F1-score
Chat	97.1	98.8	97.9	99.8	99.7	99.7
Email	98.8	97.2	98.0	99.6	99.3	99.5
File-Transfer	100	99.6	99.8	100	100	100
Streaming	99.8	99.8	99.8	98.3	99.0	98.7
VoIP	99.4	99.8	99.6	98.8	98.5	98.7
VPN-Chat	99.6	99.2	99.4	99.6	99.8	99.7
VPN-Email	99.6	100	99.8	99.0	98.9	99.0
VPN-FT	99.4	96.0	97.7	98.4	97.6	98.0
VPN-P2P	99.8	100	99.9	99.8	99.9	99.9
VPN-	99.8	99.8	99.8	99.8	99.9	99.9
Streaming	96.5	99.2	97.8	98.6	98.9	98.7
VPN-VoIP						

be compared with previous research, the 10 most recognized methodologies were chosen. For an accurate comparative experiment, the precedents that used the ISCX VPN-nonVPN dataset were chosen as comparisons, and the following methodologies were used: FlowPrint [19-20], which is a methodology that generates fingerprints for classification; CUMUL [23], AppScanner [27], and BIND [28], which extracts

and utilizes statistical properties; DeepPacket [29], FS-Net [31], and GraphDApp [32], which are based on deep learning; and ET-BERT (flow) [10], ET-BERT(packet) [10], BFCN [11], PERT [38], which applied pre-training methodology. The experiment results from the above methodologies were used for comparison. According to the analysis in Table 5, our methodology presents an excellent performance metric, and it surpasses the precedent studies in service category classification. The model proposed in this paper presents outstanding overall performance in four evaluation metrics, including the F1-score. Moreover, in this paper, only the protocol packet header information, excluding the 5-tuple, is used, while the BFCN and ET-BERT models utilized sentences consisting of 86 tokens encrypted 2-byte payloads. This leads to an additional advantage compared to the BERT-based precedents in that the classification model uses fewer sentences and is not dependent of payloads stemming from randomized encryption on the system environment.

Table 6 presents the analysis results between the method proposed in this paper and Shi et al. [11] for the performance of the classification of service types. In BFCN proposed by Shi et al. [11], alongside the research of Lin et al. [10], the BURST generator was used to extract a single session flow, and the datagrams were tokenized into adjacent 2-byte tokens to generate sentences [10-11,51]. Since this method employed by Lin et al. [10] and Shi et al. [11] splits or merges tokens into 2-byte, unique features, and semantic information of service-type traffic may be lost. The model in this paper also constructs a sentence from 19 tokens containing unique features of service-type traffic packets, while the BFCN [11] is the result of a sentence that is composed of 86 tokens from the merging of header information and payload (refer to Table 6). Fig. 3 presents a comparative analysis of the F1-score performance metrics for 11 different service types. Analyzing the experimental results in Fig. 3, it can be observed that the performance improved by 1.8 % and 1.5 % for Chat and Email service types, respectively. By contrast, BFCN [11] showed higher performance than our model by 1.1 % and 0.9 % for Streaming and VoIP service types, respectively. Analyzing the confusion matrix in Fig. 4, it can be seen that our model misclassifies Streaming and VoIP types as each other. Fig. 4 shows the confusion matrix by classification of service types. The rows correspond to the actual classes of the testing dataset, while the columns represent the label information on the prediction result of our proposed model.

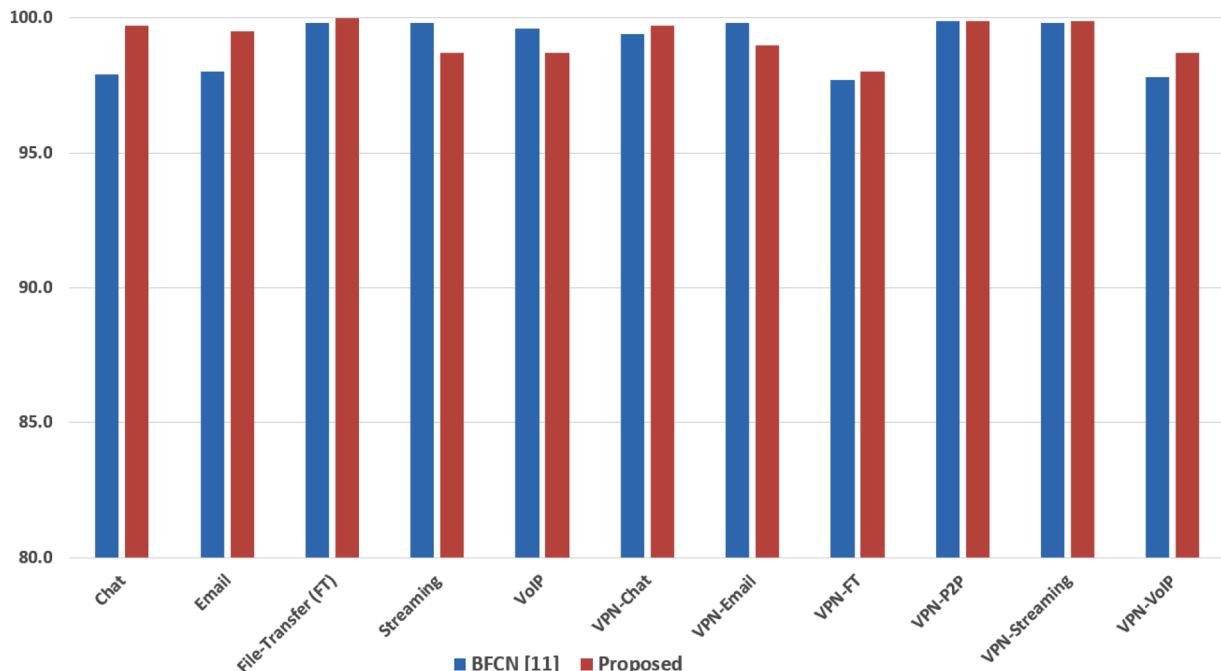


Fig. 3. Comparison of F1-score performance evaluation by service types.

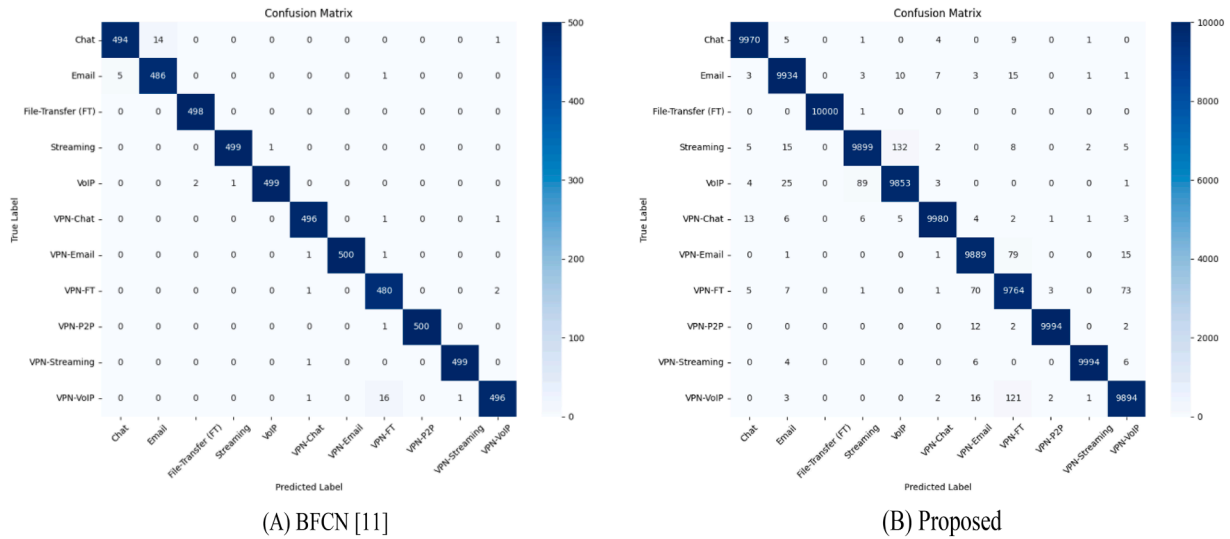


Fig. 4. Confusion matrix for service category classification.

Table 7
Comparison results on each application classification tasks.

Method	Accuracy	Precision	Recall	F1-score
AppScanner [27]	62.66	48.64	51.98	49.35
CUMUL [23]	53.65	41.29	45.35	42.36
BIND [28]	67.67	51.52	51.53	49.65
FlowPrint [19-20]	87.67	66.97	66.51	65.31
FS-Net [31]	66.47	48.19	48.48	47.37
GraphDApp [32]	63.28	59.00	54.72	55.58
DeepPacket [29]	97.58	97.85	97.45	97.65
PERT [38]	82.29	70.92	71.73	69.92
BFCN [11]	99.65	99.36	99.47	99.41
ET-BERT (flow) [10]	85.19	75.08	72.94	73.06
ET-BERT (packet) [10]	99.62	99.36	99.38	99.37
Proposed	98.74	98.76	98.73	98.74

4.4. Classification by individual applications

The second experiment involved classifying 15 application types of encrypted traffic. Previous studies [10-11, Anon., 45-46] defined and experimented with 17 types of applications. As explained in Section 4.1, the VPN-nonVPN dataset does not include the Tor and Torrent classes among the application types. Therefore, in this paper, comparative experiments were conducted while excluding the Tor and Torrent applications. 100,000 packets were randomly extracted from each application, and applications with <100,000 packets used the entirety of their data (e.g., spotify: 81,724; email: 74,144; ICQ_chat: 49,416; gmail: 48,956; AIM_chat: 42,894). In the comprehensive analysis presented in Table 7, the model proposed in this paper achieved an F1-score of 98.74 %, thus improving upon the DeepPacket [26] model based on classical deep learning 1D-CNN by 1.09 %. By contrast, when compared to the ET-BERT (packet) [10] and BFCN [11] models, the performance metric of an F1-score was reduced by 0.63 % and 0.67 %, respectively. The BFCN and ET-BERT, similar to the model proposed in this paper, excluded 5-tuple and generated 2-byte tokens from header fields and payloads to construct a sentence for pre-training and fine-tuning. However, this approach risks losing the unique characteristics of packet headers, as tokens are fixed 2-byte units derived from adjacent datagrams. For instance, the IP flags and fragment offset fields in the IP header are combined into one token, while UDP’s checksum and payload are extracted as a single token. This makes it difficult to provide clear justifications and interpretations for the classification results due to the loss of packet characteristics and semantic information. Further, in terms of practical application, there is no need for a separate BURST

Table 8
Performance evaluations of each application classification.

Application category	BFCN [11]			Proposed		
	precision	recall	F1-score	precision	recall	F1-score
AIM-Chat	99.2	97.8	98.5	99.8	98.5	99.2
Email	99.4	100	99.7	99.1	99.5	99.3
Facebook	99.2	99.0	99.1	98.0	91.7	94.7
Gmail	98.6	99.4	99.0	99.7	99.3	99.5
Hangout	100	99.6	99.8	99.4	99.5	99.4
ICQ-Chat	94.1	97.6	95.8	99.9	99.3	99.6
Netflix	100	100	100	99.7	99.8	99.7
SCP	100	99.4	99.7	100	100	100
Skype	99.4	99.8	99.6	91.9	98.3	95.0
Spotify	100	100	100	99.1	98.4	98.8
Vimeo	100	100	100	99.1	99.2	99.1
VoipBuster	100	98.8	99.4	99.4	99.9	99.6
VPN-Ftps	100	100	100	99.8	99.6	99.7
VPN-Sftp	100	99.8	99.9	99.6	99.9	99.7
Youtube	100	100	100	99.1	99.1	99.1

generator or additional modules, or any additional costs for extracting each session’s flows. Even with just 19 tokens representing the unique characteristics of the packet header, we achieved performance comparable to that presented by Lin et al. [10] and Shi et al. [11]. Finally, our BERT model guarantees classification performance without relying on dynamically changing IP address, protocol, and port numbers by excluding the 5-tuple. Moreover, by not using the randomly encrypted payloads of packet headers, we remove randomness and enhance the model’s generalization performance.

Table 8 presents the application classification performances of the proposed model in this paper, alongside the BFCN [11], and it organizes the analysis results arranged by each application. Fig. 5 compares and analyzes the F1-score performance metrics for each application between the BFCN [11] model and our model. Analyzing the experimental results in Fig. 5, it can be seen that our model outperformed the ICQ-Chat application with a 3.8 % higher F1-score. By contrast, BFCN [11] outperformed our model by 4.4 % and 4.6 % for the Facebook and Skype applications, respectively. Analyzing the confusion matrix in Fig. 6, it can be seen that packets from Facebook and Skype applications are misclassified as each other in our model. Specifically, 817 packets from the Facebook application were misclassified as Skype, while 155 packets from the Skype application were misclassified as Facebook, thus leading to a lower F1-score performance. Fig. 6 shows the confusion matrix for

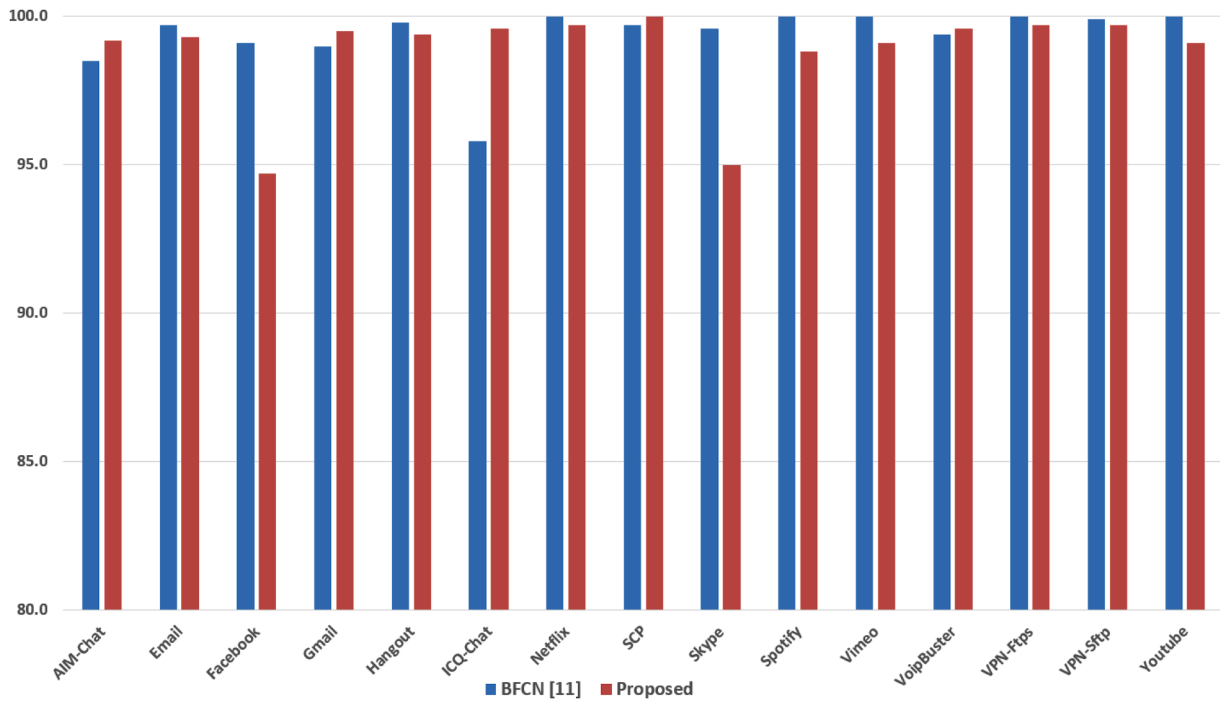


Fig. 5. Comparison of F1-score performance evaluation by application.

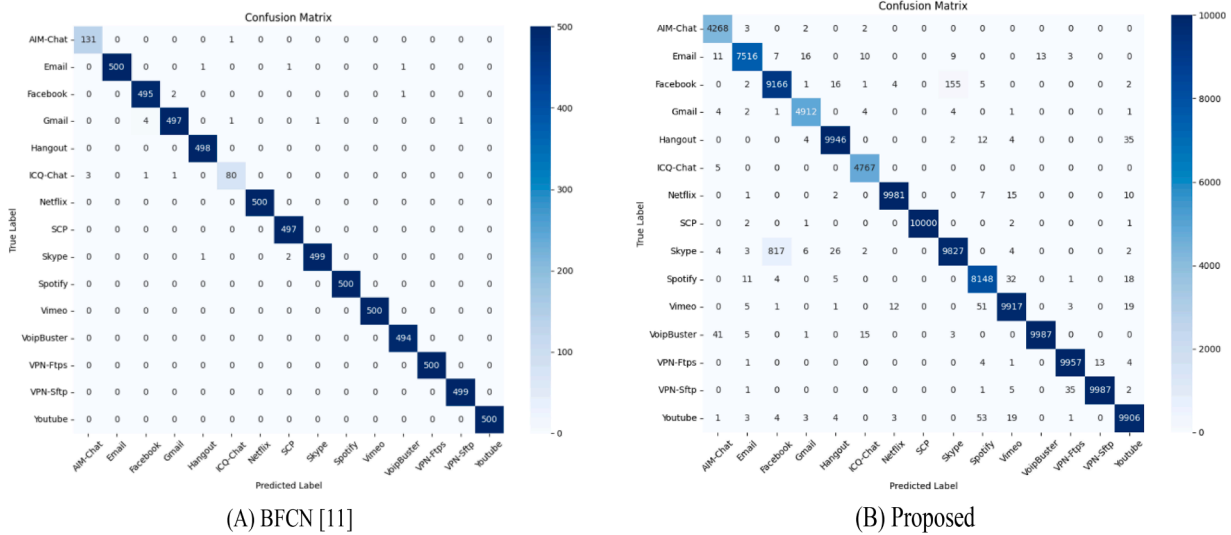


Fig. 6. Confusion matrix for each application classification.

packet classification for each application.

5. Conclusion

In this paper, we propose a novel service type and application classification system based on BERT using the packet header information of encrypted traffic. The current paper improves upon the limitations of traditional techniques and the latest deep learning methods for encrypted application classification in a number of ways: First, the proposed system extracts 2-byte tokens from packet header fields to create sentences that preserve the unique characteristics and context of the fields. These sentences are then used to classify encrypted traffic using the BERT model. By leveraging the advantages of the BERT language understanding model and performing pre-training and fine-tuning with the generated sentences, the system ensures that encrypted traffic is

accurately classified. Secondly, this paper ensures the generalization performance of the classification model and contributes to user privacy protection by using only header field information, excluding the 5-tuple and payload. Recent trends in dynamic IP allocation, port number changes, and randomly encrypted payloads have caused biases and performance degradation in classification models. Therefore, by removing the 5-tuple and payload information, this study achieves a classification model with enhanced generalization performance, thus making it applicable to real-world and changing network environments. Thirdly, the proposed system is designed such that it can be finely tuned to the classification objectives by service type and application. Using the ISCX VPN-nonVPN dataset, it achieved high performance with F1-scores of 99.24 % for service type classification and 98.74 % for application classification. The system also allows for incremental updates of the classification model through fine-tuning alone, without the need for pre-

training, even when new encrypted traffic applications are added. Lastly, the BERT-based encrypted traffic classification model proposed in this paper can be used in various ways. This capability can be utilized to maintain the confidentiality of encrypted traffic, network security monitoring, anomaly detection, support a stable network environment, Quality of Service (QoS), and traffic management in complex IT environments.

For future research directions, we will propose studies that involve selecting new features that include packet header information or signatures for faster and more accurate encrypted traffic classification. Future studies can also focus on enhancing the real-time processing capabilities of encrypted traffic, discovering useful embedded knowledge, improving model interpretability, and addressing new challenges posed by continually evolving encryption methods.

CRedit authorship contribution statement

Jaehak Yu: Writing – original draft, Visualization, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Yangseo Choi:** Writing – review & editing, Validation, Resources, Methodology, Formal analysis. **Kijong Koo:** Writing – review & editing, Visualization, Supervision, Resources, Formal analysis. **Daesung Moon:** Writing – review & editing, Validation, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

www.unb.ca/cic/datasets/vpn.html.

Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2023-00235509, Development of security monitoring technology based network behavior against encrypted cyber threats in ICT convergence environment)

References

- [1] J. Zhao, Q. Li, Y. Hong, M. Shen, MetaRockETC: Adaptive encrypted traffic classification in complex network environments via time series analysis and meta-learning, *IEEE Transactions on Network and Service Management* 21 (2) (2024) 2460–2476.
- [2] T. Liu, X. Ma, L. Liu, X. Liu, Y. Zhao, N. Hu, K.Z. Ghafour, LAMBERT: Leveraging attention mechanisms to improve the BERT fine-tuning model for encrypted traffic classification, *Mathematics* 12 (11) (2024) 1–22.
- [3] J.H. Yu, H.S. Lee, Y.H. Im, M.S. Kim, D.H. Park, Real-time classification of Internet application traffic using a hierarchical multi-class SVM, *KSI Transactions on Internet and Information Systems* 4 (5) (2010) 859–876.
- [4] G. Aceto, D. Ciunzo, A. Montieri, A. Pescape, Toward effective mobile encrypted traffic classification through deep learning, *Neurocomputing* 409 (7) (2020) 306–315.
- [5] H. Shi, H. Li, D. Zhang, C. Cheng, X. Cao, An efficient feature generation approach based on deep learning and feature selection techniques for traffic classification, *Comput. Netw.* 132 (26) (2018) 81–98.
- [6] K. Zhou, W. Wang, C. Wu, T. Hu, Practical evaluation of encrypted traffic classification based on a combined method of entropy estimation and neural networks, *ETRI Journal* 42 (3) (2020) 311–323.
- [7] M. Shafiq, X. Yu, A. Laghari, L. Yao, N. Karn, F. Abdessamia, Network traffic classification techniques and comparative analysis using machine learning algorithms, in: 2nd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 14–17 October, 2016, pp. 2451–2455.
- [8] T. Obasi, M. Shafiq, CARD-B: a stacked ensemble learning technique for classification of encrypted network traffic, *Comput. Commun.* 190 (2022) 110–125.
- [9] S. Roy, T. Shapira, Y. Shavitt, Fast and lean encrypted Internet traffic classification, *Comput. Commun.* 186 (2022) 166–173.
- [10] X. Lin, G. Xiong, G. Gou, Z. Li, J. Shi, J. Yu, Et-bert: A contextualized datagram representation with pre-training transformers for encrypted traffic classification, in: *ACM Web Conference 2022 (WWW '22)*, Lyon, France, 25–29 April, 2022, pp. 633–642.
- [11] Z. Shi, N. Luktarhan, Y. Song, G. Tian, BFCN: A novel classification method of encrypted traffic based on BERT and CNN, *Electronics (Basel)* 12 (3) (2023) 1–16.
- [12] Y. Liu, X. Wang, B. Qu, F. Zhao, ATVTISC: A novel encrypted traffic classification method based on deep learning, *IEEE Transactions on Information Forensics and Security* (2024) 1–17, <https://doi.org/10.1109/TIFS.2024.3433446>.
- [13] W. Cai, C. Hou, M. Cui, B. Wang, G. Xiong, G. Gou, Incremental encrypted traffic classification via contrastive prototype networks, *Comput. Netw.* 250 (2024) 1–12.
- [14] S. Rezaei, X. Liu, Deep learning for encrypted traffic classification: An overview, *IEEE Communications Magazine* 57 (5) (2019) 76–81.
- [15] S. Soleymannpour, H. Sadr, H. Beheshti, An efficient deep learning method for encrypted traffic classification on the web, in: 2020 6th International Conference on Web Research (ICWR), Tehran, Iran, 22–23 April, 2020, pp. 209–216.
- [16] G. Aceto, D. Ciunzo, A. Montieri, Pescape A, Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges, *IEEE Transactions on Network and Service Management* 16 (2) (2019) 445–458.
- [17] P. Velan, M. Cermak, P. Celeda, M. Drasar, A survey of methods for encrypted traffic classification and analysis, *Network Management* 25 (5) (2015) 355–374.
- [18] R.T. Elmaghraby, N.M.A. Aziem, M.A. Sobh, A.M. Bahaa-Eldin, Encrypted network traffic classification based on machine learning, *Ain Shams Engineering Journal* 15 (2) (2024) 1–10.
- [19] T.V. Ede, R. Bortolameotti, A. Continella, J. Ren, D.J. Dubois, M. Lindorfer, D. Choffnes, M.V. Steen, A. Peter, Flowprint: Semi-supervised mobile-app fingerprinting on encrypted network traffic, in: *Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, USA, 23–26 February, 2020, pp. 1–18.
- [20] J. Holland, P. Schmitt, N. Feamster, P. Mittal, New directions in automated traffic analysis, in: 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS'21), NY, USA, Association for Computing Machinery, 15–19 November, 2021, pp. 3366–3383.
- [21] P.C. Lin, Y.D. Lin, Y.C. Lai, T.H. Lee, Using string matching for deep packet inspection, *Computer (Long Beach, Calif)* 41 (4) (2008) 23–28.
- [22] Q. Ma, W. Huang, Y. Jin, J. Mao, Encrypted traffic classification based on traffic reconstruction, in: 4th International Conference on Artificial Intelligence and Big Data, Chengdu, China, 28–31 May, 2021, pp. 572–576.
- [23] A. Panchenko, F. Lanze, J. Pennekamp, T. Engel, A. Zinnen, M. Henze, K. Wehrle, Website Fingerprinting at Internet scale, in: *Network and Distributed System Security (NDSS) Symposium*, San Diego, CA, USA, 21–24 February, 2016, pp. 1–15.
- [24] T. Bujlow, V. Carela-Espanol, P. Barlet-Ros, Independent comparison of popular DPI tools for traffic classification, *Comput. Netw.* 76 (2015) 75–89.
- [25] Z. Shi, N. Luktarhan, Y. Song, H. Yin, TSFN: A novel malicious traffic classification method using BERT and LSTM, *Entropy* 25 (5) (2023) 1–15.
- [26] A. Dainotti, A. Pescape, K.C. Claffy, Issues and future directions in traffic classification, *IEEE Network* 26 (1) (2012) 35–40.
- [27] V.F. Taylor, R. Spolaor, M. Conti, I. Martinovic, Robust smartphone app identification via encrypted network traffic analysis, *IEEE Transactions on Information Forensics and Security* 13 (1) (2017) 63–78.
- [28] K. Al-Naami, S. Chandra, A. Mustafa, L. Khan, Z. Lin, K. Hamlen, B. Thuraisingham, Adaptive encrypted traffic fingerprinting with bi-directional dependence, in: *Proceedings of the 32nd Annual Conference on Computer Security Applications (ACSAC '16)*, Los Angeles, CA, USA, 5–9 December, 2016, pp. 177–188.
- [29] M. Lotfollahi, M.J. Siavoshani, R.S.H. Zade, M. Saberian, Deep packet: A novel approach for encrypted traffic classification using deep learning, *Soft. comput.* 24 (3) (2020) 1999–2012.
- [30] W. Wang, M. Zhu, J. Wang, X. Zeng, Z. Yang, End-to-end encrypted traffic classification with one-dimensional convolution neural networks, in: 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), Beijing, China, 22–24 July, 2017, pp. 43–48.
- [31] C. Liu, L. He, G. Xiong, Z. Cao, Z. Li, Fs-net: A flow sequence network for encrypted traffic classification, in: *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, Paris, France, 29 April - 2 May, 2019, pp. 1171–1179.
- [32] M. Shen, J. Zhang, L. Zhu, K. Xu, X. Du, Accurate decentralized application identification via encrypted traffic analysis using graph neural networks, *IEEE Transactions on Information Forensics and Security* 16 (2021) 2367–2380.
- [33] P. Sirinam, M. Imani, M. Juarez, M. Wright, Deep fingerprinting: Undermining website fingerprinting defenses with deep learning, in: 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS 2018), Toronto, ON, Canada, 15–19 October, 2018, pp. 1928–1943.
- [34] K. Lin, X. Xu, H. Gao, TSCRNN: A novel classification scheme of encrypted traffic based on flow spatiotemporal features for efficient management of IIoT, *Comput. Netw.* 190 (8) (2021) 1–11.
- [35] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, *ArXiv*. 2018 (2018) 1–16, <https://doi.org/10.48550/arXiv.1810.04805>.
- [36] X. Chen, P. Cong, S. Lv, A long-text classification method of Chinese news based on BERT and CNN, *IEEE Access*. 10 (2022) 34046–34057.
- [37] S. Sengupta, N. Ganguly, P. De, S. Chakraborty, Exploiting diversity in android tls implementations for mobile app traffic classification, in: *Proceedings of the World Wide Web Conference (WWW '19)*, San Francisco, CA, USA, 13–17 May, 2019, pp. 1657–1668.

- [38] H.Y. He, Z.G. Yang, X.N. Chen, PERT: Payload encoding representation from transformer for encrypted traffic classification, in: 2020 ITU Kaleidoscope: Industry-Driven Digital Transformation (ITU K), Ha Noi, Vietnam, 7-11 December, 2020, pp. 1–8.
- [39] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, ALBERT: A lite BERT for self-supervised learning of language representations, ArXiv. 2019 (2019) 1–17, <https://doi.org/10.48550/arXiv.1909.11942>.
- [40] X. Hu, C. Gu, Y. Chen, F. Wei, CBD: A deep-learning-based scheme for encrypted traffic classification with a general pre-training method, *Sensors* 21 (24) (2021) 1–18.
- [41] Tcpdump, Available at: <https://www.tcpdump.org/>.
- [42] Wireshark, Available at: <https://www.wireshark.org/>.
- [43] P. Luo, J. Chu, G. Yang, IP packet-level encrypted traffic classification using machine learning with a light weight feature engineering method, *Journal of Information Security and Applications* 75 (2023) 1–8.
- [44] G. Draper-Gil, A.H. Lashkari, M.S.I. Mamun, A.A. Ghorbani, Characterization of encrypted and vpn traffic using time related features, in: 2nd International Conference on Information Systems Security and Privacy (ICISSP), Rome, Italy, 19-21 February, 2016, pp. 407–414.
- [45] UNB, ISCX VPN 2016, Available at: <https://www.unb.ca/cic/datasets/vpn.html>.
- [46] C. Liu, W. Wang, M. Wang, F. Lv, M. Konan, An efficient instance selection algorithm to reconstruct training set for support vector machine, *Knowl. Based. Syst.* 116 (15) (2017) 58–73.
- [47] J. Yu, S. Park, S.H. Kwon, K.H. Cho, H. Lee, AI-based stroke disease prediction system using ECG and PPG bio-signals, *IEEe Access.* 10 (2022) 43623–43638.
- [48] S. Cui, B. Jiang, Z. Cai, Z. Lu, S. Liu, J. Liu, A session-packets-based encrypted traffic classification using capsule neural networks, in: 2019 IEEE 21st International Conference on High Performance Computing and Communications (HPCC/SmartCity/DSS), Zhangjiajie, China, 10-12 August, 2019, pp. 429–436.
- [49] T. Shapira, Y. Shavitt, Flowpic: Encrypted Internet traffic classification is as easy as image recognition, in: IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Paris, France, 29 April - 02 May, 2019, pp. 680–687.
- [50] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, ArXiv. 2014 (2014) 1–15, <https://doi.org/10.48550/arXiv.1412.6980>.
- [51] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, ArXiv. 2019 (2019) 1–5, <https://doi.org/10.48550/arXiv.1910.01108>.



Jaehak Yu received the B.S. degree in computer science from Konkuk University, Republic of Korea, in 2001, and the M.S. and Ph.D. degrees in computer science from Korea University, Republic of Korea, in 2003 and 2010, respectively. Since 2010, he has been with the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea, where he is currently a Principal Member with the Department of Cyber Security Research Division. His recent research interests include artificial intelligence (AI), network traffic management, network security, deep learning, and data mining.



Yangseo Choi received the B.S. degree in computer science from Kang-Won National University, Republic of Korea, in 1996, and the M.S. degrees in computer engineering from Sogang University, Republic of Korea, in 2000, and Ph.D. degrees in computer engineering from Chung-Nam National University, Republic of Korea, in 2011, respectively. Since 2000, he has been with the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea, where he is currently a Principal Member with the Department of Cyber Security Research Division. His recent research interests include network traffic analysis, vulnerability analysis, machine learning based data analysis and engineering, the CPS security, and intelligent cyber/network security.



Kijong Koo received the B.S. and M.S degree in electronics engineering from Chungnam National University, Republic of Korea, in 1999 and 2001, respectively. Since 2000, he has been with the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea, where he is currently a principal researcher in Cyber Security Research Division. His recent research interests include artificial intelligence (AI), machine learning, deep learning, reinforcement learning, autonomous penetration testing, and intelligent cyber/network security.



Daesung Moon received the M.S. degree from the Department of Computer Engineering, Busan National University, South Korea, in 2001, and the Ph.D. degree in computer science from Korea University, South Korea, in 2007. He joined the Electronics and Telecommunications Research Institute, in 2000, where he is currently a Principal Researcher. His research interests include network security, data mining, and AI Security.