Towards Context-Aware Traffic Classification via Time-Wavelet Fusion Network

Ziming Zhao
Zhejiang University
Hangzhou, China
zhaoziming@zju.edu.cn

Zhuoxue Song
Zhejiang University
Hangzhou, China
songzhuoxue@zju.edu.cn

Xiaofei Xie
Singapore Management University
Singapore, Singapore
xfxie@smu.edu.sg

Zhaoxuan Li
Institute of Information Engineering,
CAS, Beijing, China
lizhaoxuan@iie.ac.cn

Jiongchi Yu
Singapore Management University
Singapore, Singapore
jcyu.2022@phdcs.smu.edu.sg

Fan Zhang*
Zhejiang University
Hangzhou, China
fanzhang@zju.edu.cn

Tingting Li
Zhejiang University
Hangzhou, China
litt2020@zju.edu.cn

Abstract

Encrypted traffic classification occupies a significant role in cybersecurity and network management. The existing encrypted traffic classification technology mostly relies on intra-flow semantics for extracting features. However, considering that some attack behaviors inherently have similar patterns to legitimate behaviors, and powerful adversaries could simulate benign users to conceal their attack intentions, intra-flow features may be similar between different categories. In this paper, we propose TrafficScope, a time-wavelet fusion network based on Transformer to enhance the performance of encrypted traffic classification. Specifically, in addition to using intra-flow semantics, TrafficScope also extracts contextual information to construct more comprehensive representations. Moreover, to cope with the non-stationary and dynamic contextual traffic, we employ wavelet transform to extract invariant features. For feature fusion, the cross-attention mechanism is adopted to inline combine temporal and wavelet-domain features. We extensively evaluate TrafficScope compared with 7 state-of-the-art baselines based on four groups of real-world traffic datasets, the results show that TrafficScope outperforms existing methods. We conduct a series of experiments in terms of similar intra-flow feature evaluation, data pollution, flow manipulations, and dynamic context to demonstrate the robustness and stability of the proposed method. Furthermore, we produce additional experiments to present the potential of TrafficScope in cross-dataset scenarios.

CCS Concepts

• **Security and privacy → Network security**; • **Information systems → Traffic analysis**.

Keywords

Traffic Classification, Wavelet Analysis, Attention Mechanism

ACM Reference Format:
Ziming Zhao, Zhuoxue Song, Xiaofei Xie, Zhaoxuan Li, Jiongchi Yu, Fan Zhang, and Tingting Li. 2025. Towards Context-Aware Traffic Classification via Time-Wavelet Fusion Network. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3690624.3709315

KDD Availability Link:
The source code of this paper has been made publicly available at https://doi.org/10.5281/zenodo.14699123.

1 Introduction

Network traffic classification, which aims at identifying network traffic categories such as intrusion detection, applications/web services identification, malware detection, and so on, has emerged as a critical task for cyberspace security and management [1, 2].

With the widespread use of encryption protocols (*e.g.,* Secure Socket Layer/Transport Layer Security (SSL/TLS) [3, 4]), traditional deep packet inspection (DPI) such as signature-based and payload-based solutions have encountered some limitations. Therefore, adopting machine learning (ML) techniques for modeling encrypted traffic has attracted widespread attention across both academia and industry [5].

Existing ML-based traffic analysis methods can be roughly divided into two categories from the perspective of feature extraction. (i) *Packet-based* methods [6–8] directly extract relevant fields from each single raw packet, such as total length, TCP flags, time to live (TTL), *etc.* The extracted field information will be parsed and fed into the ML classifier for subsequent model inference. (ii) *Flow-based* methods analyze traffic by sessions, which are divided according

*Corresponding author.

KDD '25, Toronto, ON, Canada

ACM ISBN 979-8-4007-1245-6/25/08
https://doi.org/10.1145/3690624.3709315

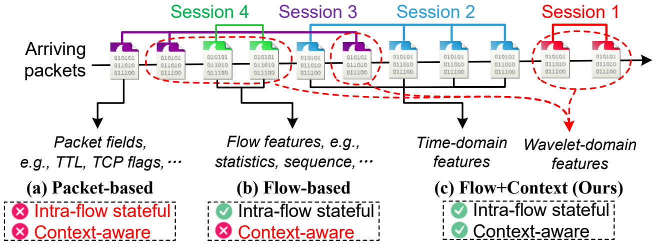


Figure 1: The illustration of different feature extractions.

to five-tuple indexes, *i.e.,* {*Source IP, Source Port, Destination IP, Destination Port, Protocol*}. Such method can learn *intra-flow* patterns, *e.g.,* calculating statistical features [9], extracting sequence information [10–12], and generating histogram distribution [13]. As shown in Figure 1, compared to flow-based methods, packet-based schemes mainly focus on detailed packet information, while failing to track flow-level state information, which leads to limited effectiveness in traffic detection [12, 14].

Although flow-based methods demonstrate some advantages, intra-flow feature representation still has certain limitations. On the one hand, some attack behaviors may natively have similar patterns to legitimate user operations. As the DDoS filtering service provider [15] reports the Empty Connection Flood attack is no different from that of a legitimate user from the intra-flow analysis perspective. Also, a POP3 mail server might get queried in the same interval as a bot communication with its C&C server, and their traffic sizes might accidentally match [16]. On the other hand, some malicious adversaries often perform several behaviors that simulate benign users to cover up their attack intentions. For instance, benign and malware applications may generate similar traffic when they use shared third-party libraries and Content Delivery Network (CDN) services [17]. These examples indicate that it is difficult to accurately identify traffic based solely on a single flow (session). More details can be found in § 2.1. This motivates us to advance *context-aware* traffic analysis, as shown in Figure 1 (c), which not only considers intra-flow semantics but also observes inter-flow information (neither packet-based nor flow-based scheme has this context-aware capability).

In this paper, we propose a time-wavelet fusion network named *TrafficScope* to enhance encrypted traffic classification by extracting features in terms of intra-flow and contextual information from the network traffic, respectively. For intra-flow features, TrafficScope focuses on the raw packet bytes of the flow. For contextual features, TrafficScope will aggregate the feature sequences of contextual traffic at different time granularities. These aggregated sequences will be conducted wavelet transform to cope with non-stationary and dynamic contexts (more explanation in § 2.2). To this end, TrafficScope is designed with three tightly coupled modules to generate traffic representations. Specifically, the extracted intra-flow features and the contextual features are separately input into the ① *Temporal Flow Representation* module and ② *Contextual Traffic Representation* module. These modules adopt the self-attention mechanism of Transformer [18] encoder blocks for modeling and mining temporal semantics and contextual information from the traffic. In the ③ *Feature Fusion* module, the temporal intra-flow representations are integrated with contextual representations with the cross-attention mechanism [19] to obtain an overall representation vector. Finally, the obtained fused representations are fed into the tailored classification layer to identify traffic categories.

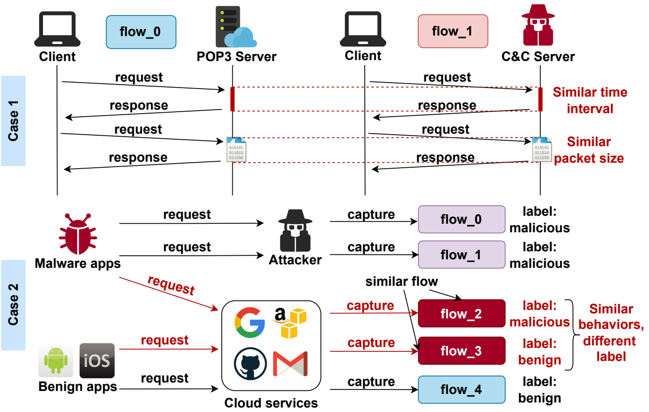


Figure 2: The illustration of similar intra-flow features.

Contributions. Our contributions can be summarized as:

- Considering that solely intra-flow features are insufficient to accurately characterize traffic, we propose enhancing the encrypted traffic representation by combining temporal intra-flow and contextual information, which offers a new perspective for analyzing encrypted traffic.
- We present leverage wavelet transform to profile contexts thus adapting to non-stationary and dynamic traffic characteristics. Meanwhile, we design a time-wavelet fusion network named TrafficScope to model intra-flow semantics and contextual information via three tightly coupled Transformer modules.
- We conduct extensive experiments on four real-world encrypted traffic datasets and results show that TrafficScope outperforms state-of-the-art methods on multiple metrics. In terms of similar intra-flow feature evaluation, data pollution, flow manipulations, and dynamic context, the results demonstrate the robustness and stability of TrafficScope. Furthermore, we produce additional experiments to present the potential of TrafficScope in cross-dataset scenarios.

2 Motivation

We outline here the motivations for leveraging the wavelet transform to develop inter-flow features, and introduce the advantages of Transformer modeling in § A.1.

2.1 Why Intra-Flow Features are Insufficient?

The existing traffic classification methods mainly extract the intra-flow features as the traffic representation. However, existing research [16] and vendor [15] reports indicate that attack and benign traffic can be very similar, indicating the intra-flow features may not be sufficient for accurate traffic characterization. (i) On the one hand, some attack behaviors may inherently have similar patterns to legitimate user operations. For instance, as the case 1 in Figure 2 shows, a POP3 mail server might get queried in the same interval as a malicious bot communication with its C&C server, and their traffic sizes might accidentally match [16]. Besides, according to the DDoS filtering service provider [15, 20, 21], the Empty Connection Flood attack is indistinguishable from that of the single three-way handshake behavior of a legitimate user. (ii) On the other hand, some malware deliberately simulates behaviors of benign users to cover up its attack intentions [22–25]. For example, as shown in case 2 of Figure 2, malware and benign applications may generate similar or even the same traffic when they use shared third-party libraries and Content Delivery Network (CDN) services [17], *e.g.,* visit common web services such as GitHub, Twitter, Google Storage, *etc.* Therefore, it is of great importance to enhance richer traffic representation beyond intra-flow features.

2.2 Why Adapt Wavelet to Construct Context?

Given the limitations of relying on intra-flow features, we aim to optimize the flow analysis with the conjunction of contextual information. However, it is non-trivial to extract representative features from contextual flows as network traffic is non-stationary [26]. In the fields of image and signal processing, the wavelet transform has demonstrated a strong capability in retaining translation- and scale-invariant properties [27, 28]. Moreover, it has been outlined in practical signal processing theory [29] that wavelet transform demonstrates theoretical superiority over Fourier transform in feature representation of non-stationary signals such as network traffic. Correspondingly, we also observe the advantages of wavelet transform in characterizing network traffic in practice. In Figure 3, the patterns of contextual flows in the time domain may vary significantly due to different runtime behaviors of applications, which involves time-shifting in subfigure (a) and diverse scales (durations) in subfigure (b). After applying wavelet transform (from left to right), we observe that the traffic contexts within the same category exhibit similar patterns in the wavelet domain, despite exhibiting variations in the time domain. Thus, we could obtain similar representations for traffic context within the same category (*e.g.,* intra-DoS or intra-benign) and different representations for contexts of different categories (*e.g.,* between benign and DoS).

3 Problem Formulation

In this section, we introduce the adversary model and assumptions, as well as provide specific problem definitions in this work.

3.1 Threat Model and Assumptions

Adversary Model. We consider strong adversaries will simulate benign user behaviors to conceal their attack intentions [22, 23, 25, 30]. This means that intra-flow features have relatively large similarities between different traffic categories. The powerful attacker may also deliberately mix mislabeled samples for data pollution. As time and environment change, traffic characteristics are not set in stone. Therefore, the problem of concept drift is within the scope of consideration. In addition, we mainly focus on *encrypted traffic analysis* in this paper, since the transmission content is increasingly being encrypted in existing networks, such as SSL/TLS and SSH. Concretely, we tend to characterize traffic behavior by portraying packet field distribution rather than analyzing transmission content, *e.g.,* TCP Payload.

Assumptions. We do not make assumptions about the traffic distribution, which means that the context information is dynamic and non-stationary, as is the case in the real world. We also do not make assumptions about the time between flows, *e.g.,* multiple attacks may be launched at the same time or may appear alternately. Furthermore, we consider the network-induced phenomena, such as packet loss, retransmission, out-of-order, *etc.* Meanwhile, we do not assume additional collaborations from other Internet entities, such as IP blacklists provided by security vendors.

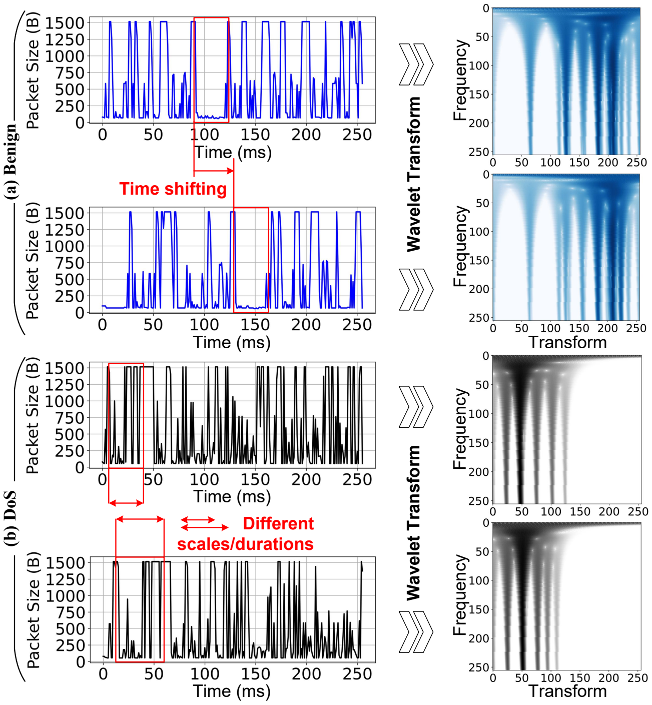


Figure 3: Wavelet transform for non-stationary context.

3.2 Problem Definition

Traffic classification refers to differentiating network traffic into different categories based on their characteristics. Specifically, our work concentrates on flow-level traffic classification. A flow in this paper is defined as a bi-directional sequence of packets that share the same 5-tuple (*i.e.,* same source/destination IP and port, same protocol). Assume that there are N samples and Q categories of traffic in total. Let the raw bytes of the i-th sample be $\mathbf{x}^{(\mathbf{i})}$. The traffic type of $\mathbf{x}^{(\mathbf{i})}$ is denoted as $y^{(i)}$, where $0 \leq y^{(i)} < Q$. We aim to build a context-aware traffic classification model $\Omega(\mathbf{x}^{(\mathbf{i})}, (\mathbf{ctx})^{(\mathbf{i})})$ to predict a label $\hat{y}^{(i)}$ that is exactly the real label $y^{(i)}$, where $(\mathbf{ctx})^{(\mathbf{i})}$ represents the contextual flows of the i-th sample.

4 Design of TrafficScope

In this section, we elaborate on the design of TrafficScope. We first present the design overview, followed by a detailed description of each module in TrafficScope.

4.1 Overview

In Figure 4, the overall pipeline refers to data pre-processing, time-wavelet fusion network, and classification layer (from left to right). As stated in § 3.2, the classification task of this paper is carried out on quintuple streams. The main difference from existing work is that our input not only includes the flow to be classified (called the flow

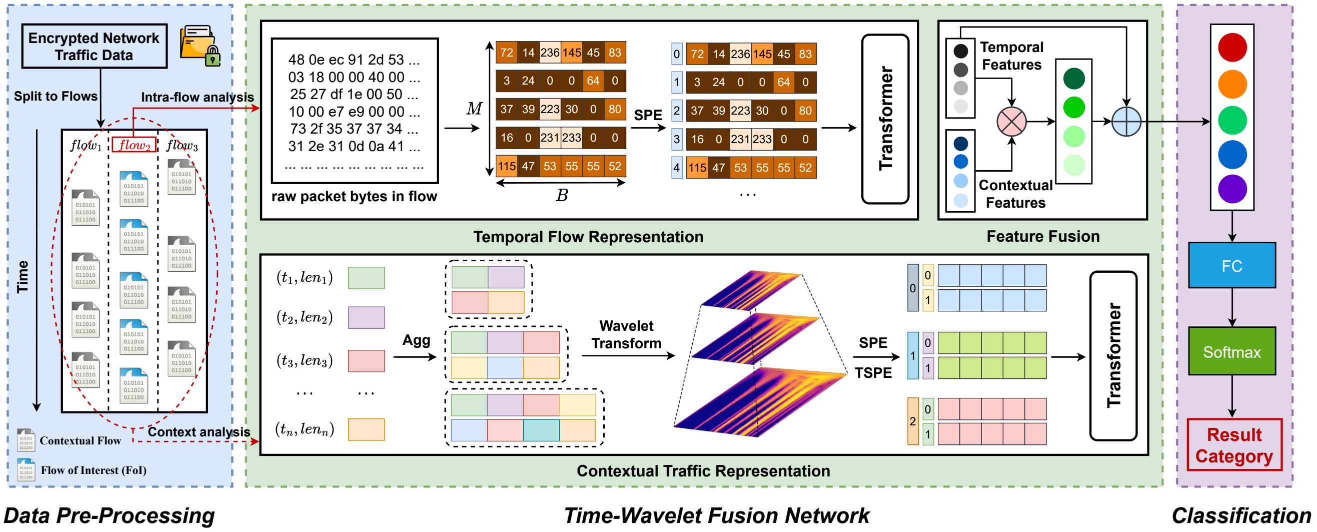


Figure 4: The overall architecture of TrafficScope.

of interest, FoI), but also involves its contextual information. The key structure of TrafficScope is the time-wavelet fusion network (the middle part of Figure 4) for traffic representation, which aims at improving encrypted traffic classification. We will elaborate on the three modules (*i.e.,* temporal flow representation, contextual traffic representation, and feature fusion) in § 4.2∼§ 4.4. Finally, we introduce the classification layer in § 4.5.

4.2 Temporal Flow Representation

Temporal flow representations are generated from raw bytes of packets in FoI. This module is designed as a general approach to extract temporal flow representation. It does not depend on the specific characteristic of the traffic, *i.e.,* the overall process of features input can be applied to both encrypted traffic and non-encrypted traffic. Besides, it should be protocol-agnostic as it is difficult to recognize and parse fields of all variant protocols [7]. Therefore, instead of performing complex packet parsing, we directly utilize raw traffic bytes as features. Specifically, we analyze the FoI, which consists of bi-directional packets with the same 5-tuple as we defined in § 3.2. We extract the first M packets from the FoI, taking the first B bytes from each packet in the flow, to form a feature matrix $\mathbf{F_t}$ in the time domain, where $\mathbf{F_t} \in \mathbb{R}^{M \times B}$ and the value range of each element in the matrix is $[0, 255]$.

We apply the sequence model Transformer [18] to characterize and capture the temporal intra-flow pattern of FoI. The packets in FoI are treated as time-series data, sorted in ascending order by timestamp, and then fed into the Transformer model. As transformers are insensitive to the order of input sequence elements, we need to perform sequence positional encoding to preserve the temporal relationships in the input sequence. We employ sequence positional encoding (SPE) of commonly used sine and cosine functions with different frequencies [18]:

$$SPE_{(i,j)} = \begin{cases} \sin\left(\frac{i}{10000^{2j/d_t}}\right), & \text{if } j \text{ is even} \\ \cos\left(\frac{i}{10000^{2(j-1)/d_t}}\right), & \text{if } j \text{ is odd} \end{cases} \tag{1}$$

where $SPE_{(i,j)}$ denotes the positional encoding value for the element (i, j) in feature matrix, d_t is the dimensionality of the temporal feature after embedding.

In summary, for the temporal feature matrix $\mathbf{F_t} = [\mathbf{m_1}, \mathbf{m_2}, \cdots, \mathbf{m_M}]$, where $\mathbf{m_i} \in \mathbb{R}^B$, we denote $\mathbf{E_t}$ as the temporal flow representation generated by the sequence model Transformer:

$$\mathbf{E_t} = Transformer\left(\left[\mathbf{W_t m_1^T}, \cdots, \mathbf{W_t m_M^T}\right] + SPE^T_{(i,j)\in\mathbf{F_t}}\right) \tag{2}$$

where $\mathbf{W_t} \in \mathbb{R}^{d_t \times B}$ is a learnable parameter.

If the number of packets in the target flow is less than M or the length of each packet is less than B, we pad the missing elements with a value of -1, which can be distinguished from the value range of normal elements. It is worth noting that the Transformer does not need a uniform sequence length. The padding here is only beneficial for data storage and batch calculation. In order to eliminate the padded values during model computation, we provide the model with a 0-1 mask matrix $\mathbf{F_{mask}}$, where 1 represents the positions that have been padded. With mask matrix, the model can identify padded positions and solely calculate original values in sequence.

With the above mechanisms, we capture the temporal patterns of FoI, generate effective temporal flow representation, and simplify the complex traffic input (encrypted or plain text, variant protocols, and packet lengths), which is used for the subsequent feature fusion.

4.3 Contextual Traffic Representation

In this module, contextual representations are generated from the contextual packet length sequence [11, 23] of FoI. To extract invariant and representative features (as stated in § 2.2) from the contextual packet length sequence, we employ the wavelet transform. The wavelet transform is a well-known time-wavelet analysis tool that has been widely used to analyze non-stationary signals and provide variant resolutions to a signal at different scales. In this paper, we utilize the wavelet transform to extract invariant and representative contextual features from the flow of interest. Mathematically, the wavelet transform used is defined as follows

$$\mathbf{W}_x(a, b) = \frac{1}{\sqrt{a}} \sum_{t=T_{start}}^{T_{end}} x(t)\Psi\left(\frac{t-b}{a}\right) \tag{3}$$

where $x(t)$ is the signal to be analyzed, $\Psi\left(\frac{t-b}{a}\right)$ is the mother wavelet function $\Psi(t)$ after scaled by a factor of a and translated by a factor of b. T_{start} and T_{end} specify the start and the end time of the signal to be processed. The wavelet coefficients $\mathbf{W}_x(a,b)$ capture the energy of the traffic signal at different scales and positions. The wavelet spectrogram can be given as

$$Spectrogram(a,b) = Norm(\log_2 ||\mathbf{W_x(a,b)}||) \tag{4}$$

where $||\cdot||$ returns the amplitude, and $Norm$ is min-max normalization function. The spectrogram efficiently represents how the instantaneous frequency changes over time. It not only provides information in the wavelet domain but also reveals the temporal variations of the signal. In this paper, we use logarithmic and normalized spectrograms as contextual features.

About contextual information, we aggregate it over a unit of time (*e.g.,* s, ms, and min). Considering the start time T_{start}, end time T_{end}, and the number of aggregation points G, we utilize hierarchical time scales for aggregation to accommodate different traffic categories. Let τ be the aggregation time scale, *i.e.,* aggregation occurs every τ seconds. If the target flow starts at time t_{FoI}, then the start time of the contextual length sequence is $T_{start} = t_{FoI} - G/2 \times \tau$, and the end time is $T_{end} = t_{FoI} + G/2 \times \tau$. That is, (**ctx**) in § 3.2 corresponds to the packets from T_{start} to T_{end}. After the aggregation of the packet length sequence, we apply wavelet transform to extract contextual features. We use the wavelet spectrum defined in Eq. (3) as the final wavelet domain feature matrix $\mathbf{F_w} \in \mathbb{R}^{C_{wt} \times G}$, where C_{wt} is the dimension of wavelet coefficients.

We apply another Transformer to characterize and capture the contextual traffic pattern of FoI. The wavelet spectrum is treated as a time series, sorted in ascending order by transform coefficients, and then fed into the Transformer model. To preserve the temporal relationships in the wavelet spectrum and the hierarchy of spectrums at different time scales, we need to perform both sequence positional encoding and hierarchical time scale positional encoding. The sequence positional encoding is the same as defined in Eq. (1). To encode information of different time scales into the sequence, we use a learnable embedding weight $\mathbf{W_{TSPE}} : N_\tau \rightarrow d_{wt}$, where N_τ represents the number of time scales used, d_{wt} is the dimensionality of the contextual feature after embedding.

In summary, for the wavelet spectrum $\mathbf{F}^\tau_{\mathbf{wt}} = [\mathbf{g_1}, \mathbf{g_2}, \cdots, \mathbf{g_G}]$, where $\mathbf{g_i} \in \mathbb{R}^{C_{wt}}$, we denote $\mathbf{E}^\tau_{\mathbf{wt}}$ as the contextual traffic representation at time scale τ generated by the sequence model Transformer:

$$\begin{aligned}\mathbf{E}^\tau_{\mathbf{wt}} = & Transformer\Big(\Big[\mathbf{W_{wt}g_1^T}, \mathbf{W_{wt}g_2^T}, \cdots, \mathbf{W_{wt}g_G^T}\Big] \\ & + SPE^T_{(i,j)\in\mathbf{F_{wt}}} + \mathbf{W_{TSPE}}idx(\tau)\Big)\end{aligned} \tag{5}$$

where $\mathbf{W_{wt}} \in \mathbb{R}^{d_{wt} \times C_{wt}}$ is a learnable parameter, $idx(\tau)$ gives the index of time scales (*i.e.,* from 1 to N_τ).

With the above mechanisms, we capture the contextual patterns of FoI, and generate effective contextual traffic representation with information on hierarchical time scales, which are used to combine with temporal features.

4.4 Time-Wavelet Feature Fusion

In this module, traffic representations are enriched by the integration of temporal flow representations with contextual traffic information. In the encrypted traffic classification task, intra-flow features could be similar between different categories. According to the motivation described in § 2, we tend to fuse temporal representations of the flow of interest (FoI) with contextual traffic representations. As depicted in Figure 5, we adopt a Transformer encoder with the cross-attention mechanism [19] to extract and integrate the relationships between temporal and contextual representations, and finally generate the fusion features. For each temporal representation vector, the cross-attention mechanism learns and calculates the weight of the contextual representation vector. The weighted sum of the contextual representation vector and the temporal representation vector is the output of the Feature Fusion module. For formalization, let $\mathbf{q_i}$, $\mathbf{k_j}$, and $\mathbf{v_j}$ be the query, key, and value vectors for the i-th element in $\mathbf{E_t}$, j-th element in $\mathbf{E_{wt}}$, respectively. The fusion representations as $\mathbf{E_f} = [\mathbf{e^1_f}, \mathbf{e^2_f}, \cdots, \mathbf{e^M_f}]$, and

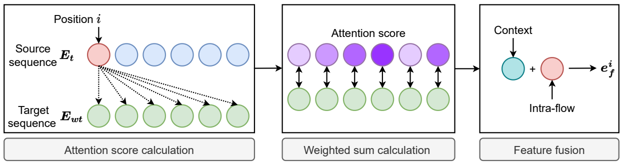


Figure 5: Feature fusion with cross-attention mechanism.

$$\mathbf{e^i_f} = Transformer\left(\mathbf{q_i} + \sum_{j=1}^{G} \frac{\mathbf{q_i} \cdot \mathbf{k_j}}{\sum_{j'=1}^{G} \mathbf{q_i} \cdot \mathbf{k_{j'}}} \cdot \mathbf{v_j}\right) \tag{6}$$

where $\mathbf{q_i} \cdot \mathbf{k_j}$ denotes the dot product between $\mathbf{q_i}$ and $\mathbf{k_j}$.

There are two main advantages to our time-wavelet fusion design. First, it provides a more comprehensive representation of traffic data. By introducing contextual information to assist the classification of FoI, such feature fusion design enables more distinguishable between different categories even if they have similar intra-flow features, thus improving the model performance. Second, the cross-attention mechanism enables the model to focus on valuable features in the contextual information for the target flow classification and cope with dynamic traffic.

4.5 Classification Layer

The task of the classification layer is to distinguish categories of FoI. After the above operations, we have obtained fusion representations for FoI. To acquire classification results, we need to learn the difference between representations of categories. So we input them into the classifier which consists of a fully connected layer with the softmax function. The output of the softmax function is the probabilities of each category. During training, the cross-entropy loss is employed to measure the difference between the predicted class probabilities and the actual labels. The cross-entropy loss as:

$$H(\mathrm{y}, \mathbf{prob}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{Q} y_{ij} \log(prob_{ij}) \tag{7}$$

where N is the number of samples, y_{ij} is the actual probability that sample i belongs to class j, and $prob_{ij}$ is the predicted probability for sample i belonging to class j. In the inference phase, the predicted category is determined as the one with the highest probability from the softmax output:

$$\hat{y_i} = \arg\max_{j=1}^{Q} prob_j \tag{8}$$

where $\hat{y_i}$ is the predicted label of the input traffic data.

Table 1: Performance comparison results (%) w.r.t. Accuracy (AC), Precision (PR), Recall (RE), and F1-Score (F1).

Datasets	**CIC-IDS2017/2018**				**CrossNet2021**				**ISCXVPN2016**				**CIC-InvesAndMal2019**			
Methods	AC	PR	RE	F1	AC	PR	RE	F1	AC	PR	RE	F1	AC	PR	RE	F1
FlowPrint [31]	86.99	90.07	86.98	87.02	82.38	89.32	88.42	87.11	79.62	80.42	78.12	78.20	72.38	71.49	71.76	72.89
FS-Net [11]	72.05	75.02	72.38	71.31	84.36	82.23	83.84	82.22	76.47	78.19	78.48	77.37	81.37	82.74	81.89	82.38
Whisper [23]	79.34	82.45	82.39	82.43	71.24	71.34	72.68	71.39	80.34	79.76	79.54	79.66	62.45	63.72	61.23	63.56
ET-BERT [32]	90.28	90.89	91.89	91.02	91.31	92.38	92.83	92.43	91.29	89.13	85.19	85.98	88.56	89.97	89.45	89.65
FlowLens [13]	91.25	88.21	90.09	87.89	83.85	82.93	84.23	83.48	86.10	78.83	86.76	86.68	84.22	84.78	84.72	84.89
HyperVision [33]	93.28	87.26	88.53	86.26	86.28	84.78	87.92	87.47	71.82	73.39	72.25	71.97	85.82	83.80	84.23	84.29
nPrint [7]	95.68	91.23	91.94	90.40	88.72	89.81	88.98	89.19	85.34	85.83	84.88	84.20	87.42	87.29	88.92	88.23
TrafficScope	**98.65**	**92.34**	**92.66**	**92.46**	**98.42**	**94.22**	**94.39**	**94.30**	**97.29**	**97.56**	**97.31**	**97.33**	**95.39**	**95.03**	**95.73**	**95.17**

5 Experiments

In this section, we perform empirical evaluations to demonstrate the effectiveness of the proposed TrafficScope framework. Specifically, we aim to answer the following research questions:

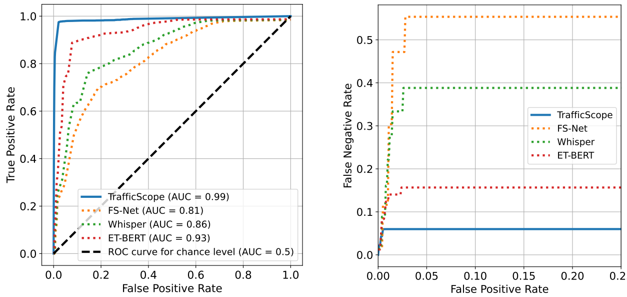
- **RQ1:** How is the traffic classification effect of TrafficScope compared to SOTA baselines? (§ 5.2)
- **RQ2:** How effective do temporal and wavelet features contribute to traffic classification? How much will the performance of TrafficScope change with different time granularity aggregation and wavelet function selection? (§ 5.3)
- **RQ3:** How does TrafficScope perform when against similar intra-flow features, data pollution, traffic manipulation, and dynamic context? (§ 5.4)
- **RQ4:** How much is the overhead of TrafficScope? (§ 5.5)
- **RQ5:** How does TrafficScope perform with cross-dataset evaluation? (§ 5.6)

5.1 Experimental Setup

Datasets. To comprehensively evaluate the effectiveness of TrafficScope, we adopt four groups of public datasets. These datasets cover common traffic classification tasks and are summarized as follows. (i) *Intrusion Detection.* CIC-IDS2017 dataset and CIC-IDS2018 dataset [34] are included in this task. (ii) *Desktop Application Identification.* CrossNet2021 dataset [35] contains traffic data from 20 categories of desktop applications such as 360, Sougou, and CSDN in two practical scenarios. (iii) *VPN Traffic Classification.* ISCXVPN2016 dataset contains pure encrypted traffic from common applications, *e.g.,* Facebook, Netflix, Skype, *etc.* (iv) *Malware Identification.* CIC-InvesAndMal2019 [36] dataset collected traffic of 426 malicious and 5065 benign applications on real smartphones. We set $train : test = 8 : 2$. More details of datasets are in § B.1.

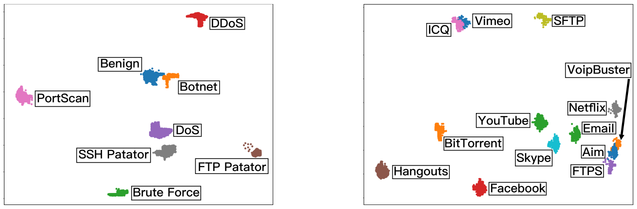
Baselines. We evaluate proposed TrafficScope framework with 7 state-of-the-art baselines of encrypted traffic classification, including FlowPrint [31], FS-Net [11], Whisper [23], ET-BERT [32], FlowLens [13], HyperVision [33], and nPrint [7]. They involve traditional machine learning, recurrent neural networks, pre-trained transformer, *etc.,* based on time-domain and frequency-domain features. More details are in § B.2.

Evaluation Metrics. To give a fair comparison, we employ evaluation metrics commonly used in traffic classification tasks, *i.e.,* Overall Accuracy (AC), Precision (PR), Recall (RE), and Macro F1-Score (F1) [37]. For four groups of datasets, all the results are reported based on multiple classification tasks.



(a) ROC Curves **(b) DET Curves**

Figure 6: Comparison results of ROC curves and DET curves.



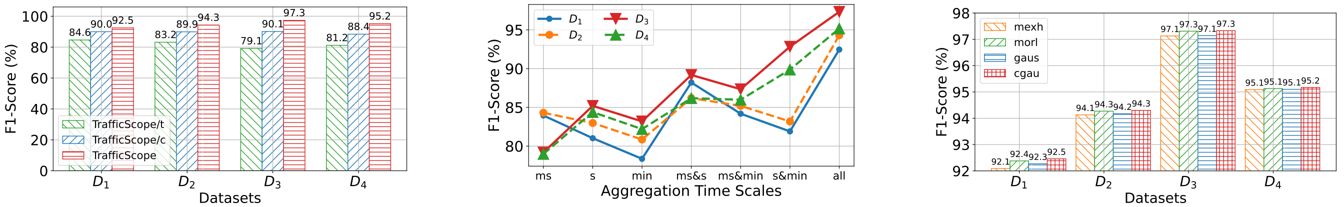
(a) Visualization of IDS dataset **(b) Visualization of VPN dataset**

Figure 7: Visualizing time-wavelet fusion features.

Implementation Details. We extract the first 64 packets of the target flow and the first 64 bytes of each packet. The dimension of the wavelet coefficients is 128. We aggregate the contextual length sequence at the millisecond level, second level, and minute level respectively. In each level, we use 128 as the number of aggregation points. The number of heads in the multi-head attention mechanism is set as 8. The number of Transformer encoder layers is set as 4. We use the dropout layer [38] with the probability of 0.5 in TrafficScope. The Adam [39] optimizer with a learning rate of 0.001 is used. TrafficScope is implemented with *PyTorch.* For all the comparison methods, we set parameters based on their official implementations. All experiments are conducted on the Ubuntu 18.04.2 server with Intel i7-12700K CPU, NVIDIA TITAN GPU, and 64GB memory.

5.2 Effectiveness Evaluation (RQ1)

Classification Performance on Various Datasets. Compared with 7 state-of-the-art (SOTA) baselines, the multiple classification results are summarized in Table 1. In terms of the four evaluation metrics, our approach outperforms other baselines by a significant margin. The most competitive baseline varies across datasets, *e.g.,* nPrint presents better performance than the other six baselines in the IDS dataset and ET-BERT dominates among baselines in

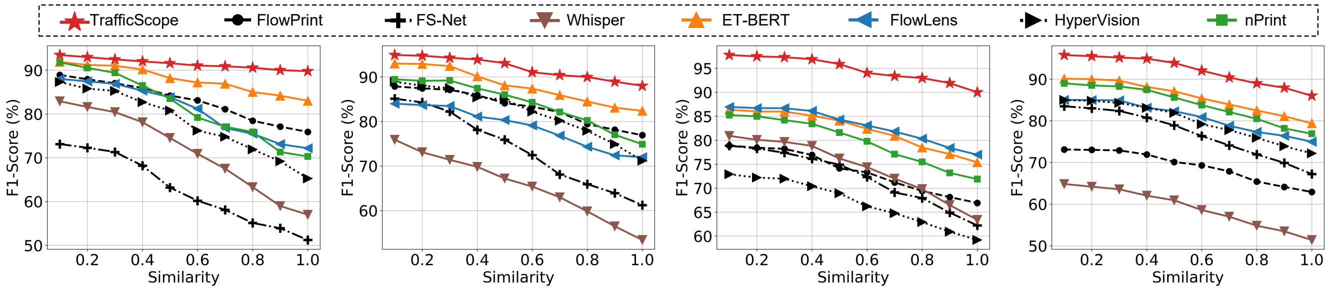


(a) Ablation study of TrafficScope

(b) Sensitivity analysis of aggregation scales

(c) Sensitivity analysis of mother wavelet functions

Figure 8: Ablation study and sensitivity analysis. D_1: IDS, D_2: CrossNet, D_3: ISCXVPN, and D_4: InvesAndMal.



(a) CIC-IDS2017/2018

(b) CrossNet2021

(c) ISCXVPN2016

(d) CIC-InvesAndMal2019

Figure 9: Performance comparison results under similar intra-flow features on four datasets.

CrossNet2021. Among these baselines, HyperVision considers the aggregation between multiple flows based on addresses. It is not robust enough because attackers will deploy botnets and change IP configuration [8, 40]. The other six baselines focus on intra-flow information, resulting in some accuracy loss.

ROC Curves and DET Curves. To further analyze the classification performance, we plot the ROC curves and the DET curves of TrafficScope and baselines (including FS-Net, Whisper, and ET-BERT that can adjust the decision threshold). From Figure 6, TrafficScope is better than other baselines in both ROC curves and DET curves. This shows the ability of TrafficScope to achieve high precision and recall while maintaining low false positive rates.

Visualization for Time-Wavelet Fusion Features. In Figure 7, we utilize t-SNE [41] to visualize time-wavelet fusion features for IDS and VPN datasets. It is clear that traffic with the same category will gather together to form clusters, which indicates the effectiveness of combining intra-flow temporal semantics and contextual wavelet information.

5.3 Ablation Study (RQ2)

In this section, we conduct the ablation study to explore the contributions of temporal and wavelet features, as well as perform experiments by using different aggregation time scales and various wavelet functions.

Feature Ablation. To explore the contributions of each feature module in TrafficScope, we eliminate the Temporal Flow Representation module (*i.e.,* TrafficScope/t) and the Contextual Traffic Representation module (*i.e.,* TrafficScope/c) respectively. As shown in Figure 8 (a), we could summarize the following observation. (i) The Temporal Flow Representation module can fully extract the temporal pattern of the flow of interest (FoI). (ii) The integration of contextual representations can more comprehensively profile the traffic and strengthen the feature representation ability. (iii) The performance of the complete model TrafficScope on the four datasets is better than other ablation models.

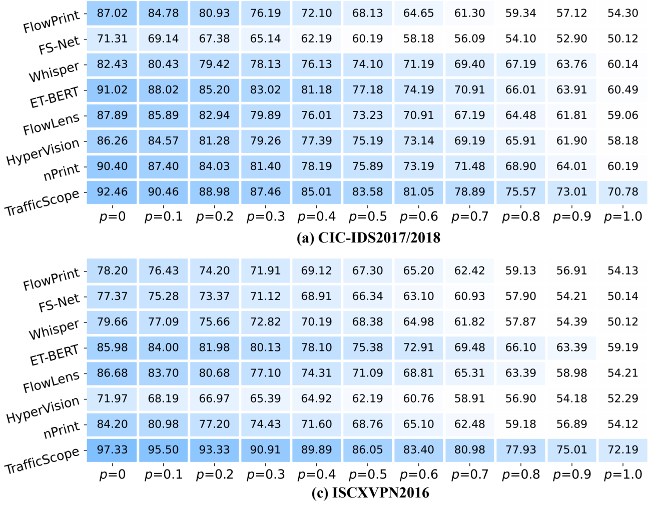
Aggregation Time Scales. From Figure 8 (b), we can find that (i) when using the three aggregation time scales together (*i.e.,* millisecond, second, and minute), the model has achieved the best performance on all datasets. (ii) In different datasets, the contributions of each time scale are various. Therefore, TrafficScope hierarchically aggregates with different time scales by default, which enhances the model's adaptability to traffic with various characteristics and ensures the effectiveness of the model.

Wavelet Functions. In Figure 8 (c) with selecting different mother wavelet functions, we can observe that *cgau* and *morl* often tend to achieve great performance. Nonetheless, with different mother wavelet function settings, the F1 score of TrafficScope does not change significantly (*e.g.,* <1%), which is still the best performance compared with other methods in Table 1. Therefore, we conclude that TrafficScope is robust to the choice of mother wavelet function and consistently outperforms other compared methods.

5.4 Robustness and Stability (RQ3)

Similar Intra-Flow Features Evaluation. We evaluate here the performance of TrafficScope and 7 compared methods, under similar intra-flow features. Details of similarity calculation are in Appendix B.3. The corresponding results are shown in Figure 9. Specifically, it is clear that as the similarity of intra-flow features increases, TrafficScope always maintains relatively stable classification performance. Therefore, our integration of temporal flow and contextual traffic representation is proved to be effective when against similar intra-flow features, echoing our original design intentions.

Data Pollution Evaluation. Furthermore, we consider data pollution with an error label of probability p in Figure 10. We observe that as the probability of error labels increases, TrafficScope maintains better performance than other baselines. This can be attributed to



	p=0	p=0.1	p=0.2	p=0.3	p=0.4	p=0.5	p=0.6	p=0.7	p=0.8	p=0.9	p=1.0
FlowPrint	87.02	84.78	80.93	76.19	72.10	68.13	64.65	61.30	59.34	57.12	54.30
FS-Net	71.31	69.14	67.38	65.14	62.19	60.19	58.18	56.09	54.10	52.90	50.12
Whisper	82.43	80.43	79.42	78.13	76.13	74.10	71.19	69.40	67.19	63.76	60.14
ET-BERT	91.02	88.02	85.20	83.02	81.18	77.18	74.19	70.91	66.01	63.91	60.49
FlowLens	87.89	85.89	82.94	79.89	76.01	73.23	70.91	67.19	64.48	61.81	59.06
HyperVision	86.26	84.57	81.28	79.26	77.39	75.19	73.14	69.19	65.91	61.90	58.18
nPrint	90.40	87.40	84.03	81.40	78.19	75.89	73.19	71.48	68.90	64.01	60.19
TrafficScope	92.46	90.46	88.98	87.46	85.01	83.58	81.05	78.89	75.57	73.01	70.78

(a) CIC-IDS2017/2018

	p=0	p=0.1	p=0.2	p=0.3	p=0.4	p=0.5	p=0.6	p=0.7	p=0.8	p=0.9	p=1.0
FlowPrint	87.11	84.43	81.11	78.91	75.12	73.30	70.10	67.42	63.13	59.91	54.10
FS-Net	82.22	78.28	75.22	72.12	67.91	65.43	62.10	59.93	55.90	53.21	50.13
Whisper	71.39	69.09	67.39	65.82	63.19	61.38	58.98	56.82	54.47	52.39	50.13
ET-BERT	92.43	88.91	85.43	82.13	79.10	76.38	72.91	69.48	66.10	63.39	61.09
FlowLens	83.48	80.67	77.48	74.10	71.31	68.09	64.81	61.31	58.39	54.98	53.29
HyperVision	87.47	84.03	81.47	78.39	74.92	71.19	68.19	64.91	60.90	56.18	52.14
nPrint	89.19	86.13	83.19	80.43	77.01	74.29	70.14	66.23	62.01	57.89	54.48
TrafficScope	94.30	92.76	90.30	87.91	85.08	83.05	81.40	78.98	75.93	73.01	70.01

(b) CrossNet2021

	p=0	p=0.1	p=0.2	p=0.3	p=0.4	p=0.5	p=0.6	p=0.7	p=0.8	p=0.9	p=1.0
FlowPrint	78.20	76.43	74.20	71.91	69.12	67.30	65.20	62.42	59.13	56.91	54.13
FS-Net	77.37	75.28	73.37	71.12	68.91	66.34	63.10	60.93	57.90	54.21	50.14
Whisper	79.66	77.09	75.66	72.82	70.19	68.38	64.98	61.82	57.87	54.39	50.12
ET-BERT	85.98	84.00	81.98	80.13	78.10	75.38	72.91	69.48	66.10	63.39	59.19
FlowLens	86.68	83.70	80.68	77.10	74.31	71.09	68.81	65.31	63.39	58.98	54.21
HyperVision	71.97	68.19	66.97	65.39	64.92	62.19	60.76	58.91	56.90	54.18	52.29
nPrint	84.20	80.98	77.20	74.43	71.60	68.76	65.10	62.48	59.18	56.89	54.12
TrafficScope	97.33	95.50	93.33	90.91	89.89	86.05	83.40	80.98	77.93	75.01	72.19

(c) ISCXVPN2016

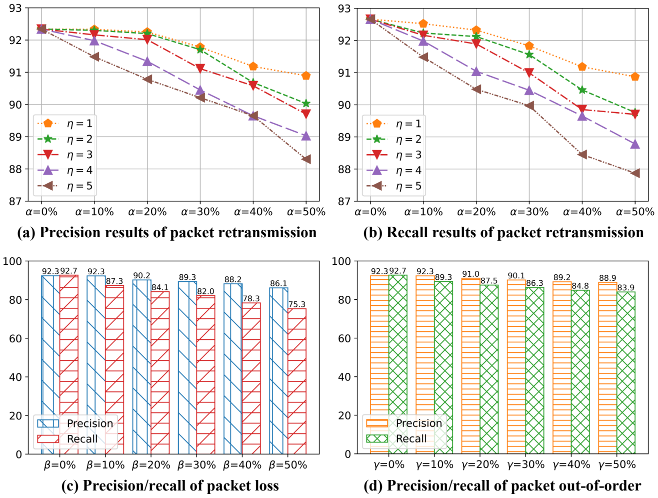
	p=0	p=0.1	p=0.2	p=0.3	p=0.4	p=0.5	p=0.6	p=0.7	p=0.8	p=0.9	p=1.0
FlowPrint	72.89	71.30	70.89	68.91	67.12	66.30	64.87	62.42	60.13	57.91	54.01
FS-Net	82.38	79.98	77.38	74.76	70.91	68.34	65.10	61.93	57.90	54.21	50.12
Whisper	63.56	62.18	61.56	60.03	58.91	57.57	56.98	55.82	54.47	52.39	50.21
ET-BERT	89.65	86.99	84.65	81.13	78.10	75.38	72.91	69.48	66.10	63.39	60.21
FlowLens	84.89	81.97	78.89	75.13	72.31	69.90	66.81	63.31	60.39	56.98	53.39
HyperVision	84.29	81.78	78.29	75.10	71.92	69.19	66.76	63.91	60.90	56.58	52.89
nPrint	88.23	85.49	82.23	79.43	76.60	73.76	70.10	67.48	64.18	59.89	54.90
TrafficScope	95.17	94.50	92.17	88.91	86.89	84.05	81.40	78.98	75.93	73.01	70.18

(d) CIC-InvesAndMal2019

Figure 10: Performance comparison results under imprecise traffic labels on four datasets.

Table 2: Dynamic context evaluation with the IDS dataset.

Mixed		κ=0	κ=1	κ=2	κ=3	κ=4	κ=5
Same	AC (%)	98.65	98.65	98.64	98.66	98.70	**98.73**
	F1 (%)	92.46	92.46	92.46	92.47	92.50	**92.84**
Diff	AC (%)	**98.65**	98.18	97.82	96.42	95.90	94.92
	F1 (%)	**92.46**	92.16	91.61	91.11	90.28	89.30



(a) Precision results of packet retransmission

(b) Recall results of packet retransmission

(c) Precision/recall of packet loss

(d) Precision/recall of packet out-of-order

Figure 11: Evaluation under manipulations on target flow.

the contextual feature that provides additional support for model classification. Therefore, TrafficScope is relatively robust against data pollution since the time-wavelet feature fusion design.

Dynamic Context Evaluation. We evaluate TrafficScope under dynamic context with the IDS dataset, by randomly selecting κ samples of contextual traffic in categories different from the FoI. As a control group, we also select κ contextual traffic of the same category as the FoI and mix them into the original contextual traffic. The experimental results are summarized in Table 2. We find that if mix the contextual traffic with the same category of FoI, the classification performance of TrafficScope could be slightly improved (*e.g.*, 98.73% accuracy in κ = 5). On the contrary, the detection results of TrafficScope decrease (~3.7% accuracy when κ = 5) when mixing multiple different types of contextual traffic. Even if mix different types of traffic into context, the performance of TrafficScope will not be greatly affected. This echoes back our intention of using wavelets to deal with non-stationary and dynamic traffic in § 2.2.

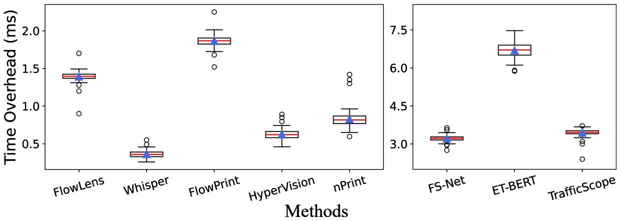


Figure 12: The time overhead.

Traffic Manipulation. To further assess the robustness of TrafficScope, we investigate the effects of various manipulations on the target flow. We mainly consider three types of real-world manipulations including packet retransmission, loss, and out-of-order of the target flows [10, 42]. Figure 11 displays the results based on the CIC-IDS2017/2018 datasets. (i) By varying the retransmission times η and probability α in subfigures (a)-(b), TrafficScope performs an average drop of 2.29% and 2.72% in precision and recall respectively, which still outperforms most baselines. (ii) By varying the packet loss probability β increases from 0% to 50% in subfigures (c), the average recall of TrafficScope drops from 92.66% to 75.29%. (iii) By varying the out-of-orde probability γ, subfigure (d) exhibits a relatively small impact than the packet loss scenario. Overall, TrafficScope achieves relatively robust performance even if occurring packet retransmission, loss, and out-of-order, given TrafficScope combines contextual information with the temporal feature for traffic classification.

5.5 Overhead Evaluation (RQ4)

We measure the time overhead in Figure 12. All models run on the Ubuntu 18.04.2 server with Intel i7-12700K CPU, NVIDIA TITAN GPU, and 64GB memory. Overall, FlowLens, Whisper, FlowPrint, HyperVision, and nPrint are on one level (<2ms) since there are machine learning based methods. Particularly, the original papers of Whisper and HyperVision use DPDK [43] for dataplane deployment,

Table 3: Running time of TrafficScope.

Steps	**Time** (*s*)
Temporal Feature Extraction	2.97×10^{-3}
Contextual Feature Extraction	5.10×10^{-4}
Temporal Transformer Forwarding	3.61×10^{-5}
Contextual Transformer Forwarding	7.73×10^{-5}
Feature Fusion	2.39×10^{-5}
Flow Classification	1.39×10^{-6}
Total	3.48×10^{-3}

Table 4: Cross-dataset evaluation.

Cross-dataset	**C17→C18**		**nonVPN→VPN**		**ScenA→ScenB**	
Metric	AC (%)	F1 (%)	AC (%)	F1 (%)	AC (%)	F1 (%)
FlowPrint	72.85	69.63	40.22	33.80	48.62	41.39
Whisper	45.21	36.88	65.03	61.82	39.84	32.68
ET-BERT	38.82	29.16	40.52	31.64	42.29	34.47
HyperVision	41.55	33.29	55.73	49.26	62.81	57.44
nPrint	32.44	25.79	38.61	29.70	44.19	31.86
TrafficScope	91.20	88.28	90.13	89.32	94.35	89.21
TrafficScope (Context)	96.91	91.07	95.73	93.69	95.02	91.61
TrafficScope (FoI)	**98.31**	**92.29**	**97.16**	**96.85**	**97.96**	**93.89**

and the time overhead here is measured based on *Python* running. The deep learning model does have more time overhead (generally than 3*ms*). The overhead of TrafficScope is close to FS-Net, ~3*ms*. Also, the most time-consuming model is ET-BERT because it contains massive parameters. In addition, we provide the overhead breakdown for TrafficScope. As Table 3, TrafficScope takes an average of $3.48 \times 10^{-3}s$ to recognize the category of the flow of interest. The main time cost is from the temporal feature extraction, which takes up $2.97 \times 10^{-3}s$.

5.6 Cross-Dataset Experiments (RQ5)

For three cross-dataset scenarios (based on IDS, VPN, and CrossNet) in Table 4, and it is clear that TrafficScope outperforms the baseline by >15% accuracy and F1 score. Meanwhile, we conduct ablation experiments on the drift of content and FoI. The results show that when only FoI drifts or context drifts, the performance of TrafficScope is only slightly affected. In real scenarios, the occurrence of concept drift is gradual, the background traffic and FoI may not change suddenly simultaneously. Overall, TrafficScope performs limited accuracy loss against cross-dataset tests.

6 Discussion

Practicality. In addition to typical intrusion detections, TrafficScope can be extended to more application scenarios. Considering the advanced persistent threat (APT) attacks usually include multiple stages [44, 45], *e.g.,* MITRE ATT&CK [46] kill chain shows that attackers often use a series of related behaviors dispersed in multiple sessions to coordinate to achieve attack goals.

Extensibility. On the one hand, applying TrafficScope for unknown detection is feasible. We could use the fusion features (the hidden layer state output by feature fusion Transformer in TrafficScope) and combine anomaly detection models [47, 48] to develop unknown attack detection. On the other hand, TrafficScope could provide a new perspective to cope with low-quality datasets [49], by considering inter-stream contextual semantics.

Limitations and Future Work. (i) The basic classification unit of TrafficScope is a flow. This implies that TrafficScope may be unsuitable for scenarios where network flows are inseparable, such as Tor. (ii) In the future, we can consider combining the latest data-plane primitives (*e.g.,* Intel DPDK [8, 23] and P4 in programmable switches [50, 51]) with TrafficScope to implement real-time traffic analysis. (iii) In the real world, traffic categories are continuously increasing [52], class-incremental learning is a promising direction.

7 Related Work

Encrypted Traffic Classification. Encrypted traffic classification [53, 54] is an essential task in network security and management [55, 56]. There are some works that leverage the attention mechanism and Transformer for traffic classification tasks [57–59]. However, PEAN [57] and YaTC [58] focus on intra-flow features (involving packet-level and flow-level), and MT-FlowFormer [59] only considers randomly introducing an additional flow for feature enhancement. These works rarely use the contextual information of flows, so they could not maintain stable performance when intra-flow features are similar.

App Fingerprinting. App fingerprinting aims to identify specific applications or services by analyzing the characteristics of their generated network traffic [31, 60, 61]. Compared with these app fingerprinting approaches, TrafficScope achieves a more general classification of encrypted traffic. Our model does not depend on specific characteristics of application traffic and is protocol-agnostic.

ML-Based NIDS. Network Intrusion Detection Systems (NIDSes) are critical components in safeguarding network infrastructures against various malicious activities [30, 62]. Numerous studies have proposed ML-based NIDS [9, 13, 23, 33]. Our work covers malicious traffic detection and can be used for NIDS in practice.

8 Conclusions

In this work, we propose TrafficScope, a time-wavelet fusion network based on Transformer to enhance the performance of encrypted traffic classification against similar intra-flow features. TrafficScope first captures the temporal relationship of packet bytes in flow and applies wavelet transform for generating invariant and representative contextual features, to realize a powerful representation ability. Extensive experiments indicate that TrafficScope's performance and practicality outperform state-of-the-art algorithms by significant margins. We believe TrafficScope provides new perspectives for encrypted traffic classification.

Acknowledgments

This work was supported in part by National Key R&D Program of China (2023YFB3106800), by the National Research Foundation, Singapore, and the Cyber Security Agency under its National Cybersecurity R&D Programme (NCRP25-P04-TAICeN), by National Natural Science Foundation of China (62227805, 62072398, 62172405), by the Natural Science Foundation of Jiangsu Province (BK20220075), by the Fok Ying-Tung Education Foundation for Young Teachers in the Higher Education Institutions of China (No.20193218210004), and by Key R&D Program of Zhejiang Province (2023C01039). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore and Cyber Security Agency of Singapore.

References

[1] Zhaoxuan Li, Ziming Zhao, Rui Zhang, et al. metanet: Interpretable unknown mobile malware identification with a novel meta-features mining algorithm. *Computer Networks*, 250:110563, 2024.

[2] Hongtao Shi, Hongping Li, Dan Zhang, Chaqiu Cheng, and Xuanxuan Cao. An efficient feature generation approach based on deep learning and feature selection techniques for traffic classification. *Computer Networks*, 132:81–98, 2018.

[3] Alan Freier, Philip Karlton, and Paul Kocher. The secure sockets layer (ssl) protocol version 3.0. Technical report, 2011.

[4] Tim Dierks and Eric Rescorla. The transport layer security (tls) protocol version 1.2. Technical report, 2008.

[5] Ly Vu, Hoang V Thuy, Quang Uy Nguyen, Tran N Ngoc, Diep N Nguyen, Dinh Thai Hoang, and Eryk Dutkiewicz. Time series analysis for encrypted traffic classification: A deep learning approach. In *2018 18th International Symposium on Communications and Information Technologies (ISCIT)*, pages 121–126. IEEE, 2018.

[6] Guorui Xie, Qing Li, Yutao Dong, et al. Mousika: Enable General In-Network Intelligence in Programmable Switches by Knowledge Distillation. In *INFOCOM*. IEEE, 2022.

[7] Jordan Holland, Paul Schmitt, Nick Feamster, and Prateek Mittal. New directions in automated traffic analysis. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 3366–3383, 2021.

[8] Ziming Zhao, Zhuotao Liu, Huan Chen, Fan Zhang, Zhuoxue Song, and Zhaoxuan Li. Effective DDoS Mitigation via ML-Driven In-network Traffic Shaping. *IEEE Transactions on Dependable and Secure Computing*, 2024.

[9] Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. Kitsune: an ensemble of autoencoders for online network intrusion detection. *arXiv preprint arXiv:1802.09089*, 2018.

[10] Ziming Zhao, Zhaoxuan Li, Jialun Jiang, Fengyuan Yu, Fan Zhang, Congyuan Xu, Xinjie Zhao, Rui Zhang, and Shize Guo. ERNN: Error-resilient RNN for encrypted traffic detection towards network-induced phenomena. *IEEE Transactions on Dependable and Secure Computing*, 2023.

[11] Chang Liu, Longtao He, Gang Xiong, Zigang Cao, and Zhen Li. FS-Net: A flow sequence network for encrypted traffic classification. In *IEEE INFOCOM 2019-IEEE Conference On Computer Communications*, pages 1171–1179. IEEE, 2019.

[12] Guangmeng Zhou and Zhuotao Liu and Chuanpu Fu and Qi Li and Ke Xu. An efficient design of intelligent network data plane. In *USENIX Security Symposium*, pages 6203–6220. USENIX Association, 2023.

[13] Diogo Barradas, Nuno Santos, Luís Rodrigues, Salvatore Signorello, Fernando MV Ramos, and André Madeira. FlowLens: Enabling efficient flow classification for ml-based network security applications. In *NDSS*, 2021.

[14] Aristide Tanyi-Jong Akem, Michele Gucciardo, and Marco Fiore. Flowrest: Practical flow-level inference in programmable switches with random forests. In *INFOCOM*, pages 1–10. IEEE, 2023.

[15] MAZEBOLT. Mazebolt knowledge base. https://kb.mazebolt.com/, 2016.

[16] Florian Tegeler, Xiaoming Fu, Giovanni Vigna, and Christopher Kruegel. BotFinder: Finding bots in network traffic without deep packet inspection. In *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, pages 349–360, 2012.

[17] Yael Daihes, Hen Tzaban, Asaf Nadler, and Asaf Shabtai. MORTON: detection of malicious routines in large-scale DNS traffic. In *Computer Security–ESORICS 2021: 26th European Symposium on Research in Computer Security, Darmstadt, Germany, October 4–8, 2021, Proceedings, Part I 26*, pages 736–756. Springer, 2021.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[19] Xi Wei, Tianzhu Zhang, Yan Li, Yongdong Zhang, and Feng Wu. Multi-modality cross attention network for image and sentence matching. In *CVPR*, pages 10938–10947. Computer Vision Foundation / IEEE, 2020.

[20] Xi Ling, Jiongchi Yu, et al. Ddosminer: An automated framework for ddos attack characterization and vulnerability mining. In *International Conference on Applied Cryptography and Network Security*, pages 283–309. Springer, 2024.

[21] Ziming Zhao, Zhaoxuan Li, Zhihao Zhou, Jiongchi Yu, Zhuoxue Song, Xiaofei Xie, Fan Zhang, and Rui Zhang. Ddos family: A novel perspective for massive types of ddos attacks. *Computers & Security*, 138:103663, 2024.

[22] Carlos Novo and Ricardo Morla. Flow-based detection and proxy-based evasion of encrypted malware C2 traffic. In *AISec@CCS*, pages 83–91. ACM, 2020.

[23] Chuanpu Fu, Qi Li, Meng Shen, and Ke Xu. Realtime robust malicious traffic detection via frequency domain analysis. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 3431–3446, 2021.

[24] Ziming Zhao, Zhaoxuan Li, Tingting Li, and Fan Zhang. Tpe-det: A tamper-proof external detector via hardware traces analysis against iot malware. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(11):3455–3466, 2024.

[25] Chuanpu Fu and Qi Li and Meng Shen and Ke Xu. Frequency domain feature based robust malicious traffic detection. *IEEE/ACM Trans. Netw.*, 31(1):452–467, 2023.

[26] Blake Anderson and David McGrew. Machine learning for encrypted malware traffic classification: accounting for noisy labels and non-stationarity. In *Proceedings of the 23rd ACM SIGKDD International Conference on knowledge discovery and data mining*, pages 1723–1732, 2017.

[27] Michael Unser and Dimitri Van De Ville. Wavelet steerability and the higher-order riesz transform. *IEEE Trans. Image Process.*, 19(3):636–652, 2010.

[28] Huilin Xiong, Tianxu Zhang, and Y. S. Moon. A translation- and scale-invariant adaptive wavelet transform. *IEEE Trans. Image Process.*, 9(12):2100–2108, 2000.

[29] Christopher Torrence and Gilbert P Compo. A practical guide to wavelet analysis. *Bulletin of the American Meteorological society*, 79(1):61–78, 1998.

[30] Karel Bartos, Michal Sofka, and Vojtech Franc. Optimized invariant representation of network traffic for detecting unseen malware variants. In *USENIX security symposium*, pages 807–822, 2016.

[31] Thijs Van Ede, Riccardo Bortolameotti, Andrea Continella, Jingjing Ren, Daniel J Dubois, Martina Lindorfer, David Choffnes, Maarten van Steen, and Andreas Peter. FLOWPRINT: Semi-supervised mobile-app fingerprinting on encrypted network traffic. In *Network and Distributed System Security Symposium (NDSS)*, volume 27, 2020.

[32] Xinjie Lin, Gang Xiong, Gaopeng Gou, Zhen Li, Junzheng Shi, and Jing Yu. ET-BERT: A contextualized datagram representation with pre-training transformers for encrypted traffic classification. In *Proceedings of the ACM Web Conference 2022*, pages 633–642, 2022.

[33] Chuanpu Fu, Qi Li, and Ke Xu. Detecting unknown encrypted malicious traffic in real time via flow interaction graph analysis. In *NDSS*. The Internet Society, 2023.

[34] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSP*, 1:108–116, 2018.

[35] Wenhao Li, Xiao-Yu Zhang, Huaifeng Bao, Haichao Shi, and Qiang Wang. Prograph: Robust network traffic identification with graph propagation. *IEEE/ACM Trans. Netw.*, pages 1–15, 2022.

[36] Laya Taheri, Andi Fitriah Abdul Kadir, and Arash Habibi Lashkari. Extensible android malware detection and family classification using network-flows and API-calls. In *2019 International Carnahan Conference on Security Technology (ICCST)*, pages 1–8. IEEE, 2019.

[37] Wenbo Zheng, Chao Gou, Lan Yan, and Shaocong Mo. Learning to classify: A flow-based relation network for encrypted traffic classification. In *Proceedings of The Web Conference 2020*, pages 13–22, 2020.

[38] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[39] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[40] Ziming Zhao, Zhaoxuan Li, Fan Zhang, Tingting Li, and Jianwei Yin. Poster: Combine topology and traffic to calibrate p2p botnet identification in large-scale network. In *Proceedings of the ACM SIGCOMM 2024 Conference: Posters and Demos*, pages 16–18, 2024.

[41] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[42] Renjie Xie, Jiahao Cao, Enhuan Dong, Mingwei Xu, Kun Sun, Qi Li, Licheng Shen, and Menghao Zhang. Rosetta: Enabling robust TLS encrypted traffic classification in diverse network environments with tcp-aware traffic augmentation. In *USENIX Security Symposium*, pages 625–642. USENIX Association, 2023.

[43] D. Project. Dpdk: Data plane development kit. [EB/OL], 2010. http://dpdk.org/ Accessed November 27, 2020.

[44] Florian Wilkens, Felix Ortmann, Steffen Haas, Matthias Vallentin, and Mathias Fischer. Multi-stage attack detection via kill chain state machines. In *CYSARM@CCS*, pages 13–24. ACM, 2021.

[45] Chunlin Xiong, Tiantian Zhu, Weihao Dong, Linqi Ruan, Runqing Yang, Yueqiang Cheng, Yan Chen, Shuai Cheng, and Xutong Chen. Conan: A practical real-time APT detection system with high accuracy and efficiency. *IEEE Trans. Dependable Secur. Comput.*, 19(1):551–565, 2022.

[46] Blake E. Strom, Andy Applebaum, Doug P. Miller, et al. Mitre att&ck: Design and philosophy. In *Technical report*. The MITRE Corporation, 2018.

[47] Ziming Zhao, Zhaoxuan Li, Xiaofei Xie, Jiongchi Yu, Fan Zhang, Rui Zhang, Binbin Chen, Xiangyang Luo, Ming Hu, and Wenrui Ma. FOSS: Towards fine-grained unknown class detection against the open-set attack spectrum with variable legitimate traffic. *IEEE/ACM Transactions on Networking*, 2024.

[48] Ziming Zhao, Zhaoxuan Li, Zhuoxue Song, et al. Trident: A universal framework for fine-grained and class-incremental unknown traffic detection. In *Proceedings of the ACM on Web Conference 2024*, pages 1608–1619, 2024.

[49] Yuqi Qing, Qilei Yin, Xinhao Deng, Yihao Chen, Zhuotao Liu, Kun Sun, Ke Xu, Jia Zhang, and Qi Li. Low-quality training data only? A robust framework for detecting encrypted malicious network traffic. *CoRR*, abs/2309.04798, 2023.

[50] Ziming Zhao, Zhaoxuan Li, Zhuoxue Song, and Fan Zhang. Work-in-progress: Towards real-time IDS via RNN and programmable switches co-designed approach. In *RTSS*, pages 431–434. IEEE, 2023.

[51] Ziming Zhao, Zhaoxuan Li, Zhuoxue Song, Fan Zhang, and Binbin Chen. Rids: Towards advanced ids via rnn model and programmable switches co-designed approaches. In *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*, pages 591–600. IEEE, 2024.
[52] Zhuoxue Song, Ziming Zhao, Fan Zhang, Gang Xiong, Guang Cheng, Xinjie Zhao, Shize Guo, and Binbin Chen. I^{2}RNN: An incremental and interpretable recurrent neural network for encrypted traffic classification. *IEEE Transactions on Dependable and Secure Computing*, 2023.
[53] Manish Marwah and Martin F. Arlitt. Deep learning for network traffic data. In *KDD*, pages 4804–4805. ACM, 2022.
[54] Kevin Fauvel, Fuxing Chen, and Dario Rossi. A lightweight, efficient and explainable-by-design convolutional neural network for internet traffic classification. In *KDD*, pages 4013–4023. ACM, 2023.
[55] Ziming Zhao, Zhaoxuan Li, Jiongchi Yu, Fan Zhang, Xiaofei Xie, Haitao Xu, and Binbin Chen. Cmd: co-analyzed iot malware detection and forensics via network and hardware domains. *IEEE Transactions on Mobile Computing*, 2023.
[56] Haozhen Zhang, Le Yu, Xi Xiao, Qing Li, Francesco Mercaldo, Xiapu Luo, and Qixu Liu. Tfe-gnn: A temporal fusion encoder using graph neural networks for fine-grained encrypted traffic classification. In *Proceedings of the ACM Web Conference 2023*, pages 2066–2075, 2023.
[57] Peng Lin, Kejiang Ye, Yishen Hu, Yanying Lin, and Cheng-Zhong Xu. A novel multimodal deep learning framework for encrypted traffic classification. *IEEE/ACM Transactions on Networking*, 31(3):1369–1384, 2022.
[58] Ruijie Zhao, Mingwei Zhan, Xianwen Deng, Yanhao Wang, Yijun Wang, Guan Gui, and Zhi Xue. Yet another traffic classifier: A masked autoencoder based traffic transformer with multi-level flow representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5420–5427, 2023.
[59] Ruijie Zhao, Xianwen Deng, Zhicong Yan, Jun Ma, Zhi Xue, and Yijun Wang. Mt-flowformer: A semi-supervised flow transformer for encrypted traffic classification. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2576–2584, 2022.
[60] Jianfeng Li, Shuohan Wu, Hao Zhou, Xiapu Luo, Ting Wang, Yangyang Liu, and Xiaobo Ma. Packet-level open-world app fingerprinting on wireless traffic. In *The 2022 Network and Distributed System Security Symposium (NDSS'22)*, 2022.
[61] Sanghak Oh, Minwook Lee, Hyunwoo Lee, Elisa Bertino, and Hyoungshick Kim. Appsniffer: Towards robust mobile app fingerprinting against vpn. In *Proceedings of the ACM Web Conference 2023*, pages 2318–2328, 2023.
[62] Liang Li, Yuanhui He, Feiyang Huang, et al. An automated alert cross-verification system with graph neural networks for ids events. In *2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 2240–2245. IEEE, 2024.
[63] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pretraining of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
[64] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.
[65] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
[66] Linhao Dong, Shuang Xu, and Bo Xu. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5884–5888. IEEE, 2018.
[67] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.
[68] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: BERT pre-training of image Transformers. *arXiv preprint arXiv:2106.08254*, 2021.
[69] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022.

A Additional Details of Transformer

A.1 Why Use the Transformer Modeling?

Transformer [18] is a neural network architecture that has shown remarkable performance in various tasks such as natural language processing [63–65], speech processing [66, 67], and computer vision [68, 69]. This motivates us to leverage the power of the Transformer for network traffic classification. The basic building block of Transformer is the self-attention mechanism, which computes the importance of different parts of a sequence based on their relevance to other parts of the same sequence. The self-attention mechanism enables the network to focus on different parts of the input sequence based on their relevance to the given task. As we visualized the attention weights of samples from multiple categories in Figure 13, the attention positions of different samples belonging to the same categories exhibit similar distributions, while the samples of different categories hold various distributions. This fact reflects that the attention mechanism can help distinguish categories by focusing on salient positions. Besides, the Transformer also includes other components such as multi-head attention and position-wise feed-forward networks, and the details are illustrated in § A. We design TrafficScope with three Transformer encoders. Among them, two encoders are fed with the original raw bytes of network traffic and wavelet spectrogram respectively, to perform linear projection and self-attention mechanism. The third Transformer encoder combines information from both the time domain and wavelet domain via the cross-attention mechanism [19]. Overall, leveraging the Transformer to model time-wavelet features allows us to extract informative representations for intra-flow and traffic context. Meanwhile, the cross-attention mechanism can construct inline fusion for features, which can be helpful to focus on significant differences in various categories.

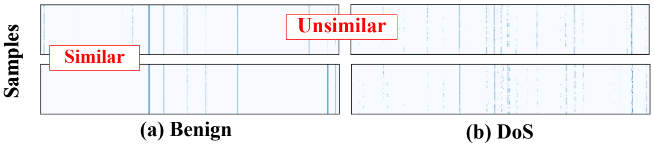


Figure 13: Visualization of the attention weight for samples.

A.2 Attention Mechanism

Self-Attention Mechanism. Suppose the input sequence is with a size of $T \times D$, representing the temporal sequence length and feature dimensions, respectively. Each self-attention head computes $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{T\times H}$ by a linear transformation, representing Query Matrix, Key Matrix, and Value Matrix, respectively.

$$\mathbf{Q} = \mathbf{X}\mathbf{W}^{\mathbf{Q}}, \mathbf{K} = \mathbf{X}\mathbf{W}^{\mathbf{K}}, \mathbf{V} = \mathbf{X}\mathbf{W}^{\mathbf{V}} \tag{9}$$

where $\mathbf{W}^{\mathbf{Q}}, \mathbf{W}^{\mathbf{K}}, \mathbf{W}^{\mathbf{V}} \in \mathbb{R}^{D\times H}$ are trainable parameters, H is the hidden layer dimension. Then the output of this self-attention head is computed as

$$\text{Self-Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^{\mathbf{T}}}{\sqrt{H}})\mathbf{V} \tag{10}$$

For each query vector $\mathbf{q_i}$, the self-attention mechanism calculates its similarity with each key vector $\mathbf{k_i}$ as weights, to perform a weighted sum on all value vectors $\mathbf{v_i}$. Therefore, the length of the output sequence length only depends on the number of query vectors. In other words, the Transformer is capable of handling sequence data of variable length without requiring explicit specification.

Multi-Head Mechanism. Multi-head attention allows the network to attend to different aspects of the input sequence in parallel, by computing multiple sets of attention weights with different linear projections of the input vectors.

$$\text{Multi-head}(\mathbf{Q},\mathbf{K},\mathbf{V}) = Concat(head_1, ..., head_h)\mathbf{W}^{\mathbf{O}} \tag{11}$$

where $head_i$ = Self-Attention$(\mathbf{X}\mathbf{W}_{\mathbf{i}}^{\mathbf{Q}}, \mathbf{X}\mathbf{W}_{\mathbf{i}}^{\mathbf{K}}, \mathbf{X}\mathbf{W}_{\mathbf{i}}^{\mathbf{V}})$. $\mathbf{W}_{\mathbf{i}}^{\mathbf{Q}} \in \mathbb{R}^{D\times H_k}$, $\mathbf{W}_{\mathbf{i}}^{\mathbf{K}} \in \mathbb{R}^{D\times H_k}$, $\mathbf{W}_{\mathbf{i}}^{\mathbf{V}} \in \mathbb{R}^{D\times H_v}$, and $\mathbf{W}^{\mathbf{O}} \in \mathbb{R}^{H_v h\times D}$ are projection parameters. The parameter h denotes the number of self-attention layers (heads). Typically, $H_k = H_v = D/h$.

Position-Wise Feed-Forward Networks. Position-wise feed-forward networks are employed to apply a non-linear transformation to the output of the self-attention mechanism.

$$FFN(x) = max(0, x\mathbf{W}_1 + b_1)\mathbf{W}_2 + b_2 \tag{12}$$

B Additional Details of Evaluations

B.1 Additional Details of Datasets

- *Intrusion Detection.* Malicious traffic identification is to recognize various encrypted attack traffic from benign traffic. CIC-IDS2017 dataset and CIC-IDS2018 dataset [34] are included in this task. The CIC-IDS2017 dataset collects network traffic across five days, including benign and a series of attacks such as Botnet, DDoS, Patator, *etc.* The CIC-IDS2018 dataset is an upgrade to the 2017 version dataset. It is more diverse and comprehensive and is based on user profiles that contain abstract representations of events and behaviors on the network.
- *Desktop Application Identification.* Application identification aims to classify encrypted application traffic into specific application categories. CrossNet2021 dataset [35] contains traffic data from 20 categories of desktop applications such as 360, Sougou, and CSDN in two practical scenarios, *i.e.,* stable (ScenarioA) and production (ScenarioB) networks. The traffic was captured using *tcpdump*, including 2.5GB of data. Note that the same categories of applications in two scenarios with different network quality-of-service (QoS), such as various bandwidths and channel disturbance, can be used to evaluate the model robustness in cross-network identification tasks.
- *VPN Traffic Classification.* As Virtual Private Networks (VPNs) are popular for bypassing censorship as well as accessing geo-locked services, which are difficult to detect due to their protocol obfuscation, we include the ISCXVPN2016 dataset in the experiment. It contains pure encrypted traffic from common applications, *e.g.,* Facebook, Netflix, Skype, *etc.* The applications are encrypted with various security protocols, including HTTPS, SSH, and proprietary protocols.
- *Malware Identification.* The CIC-InvesAndMal2019 [36] dataset collected traffic of 426 malicious and 5065 benign applications on real smartphones. The traffic types can be divided into five categories including Benign, Adware, Ransomware, Scareware, and SMS Malware.

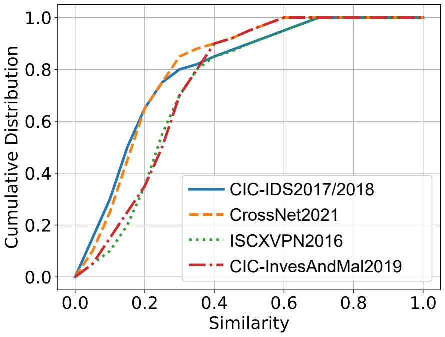


Figure 14: Cumulative distribution of similar intra-flow features on CIC-IDS2017/2018, CrossNet2021, ISCXVPN2016, and CIC-InvesAndMal2019 datasets.

B.2 Additional Details of Baselines

- **FlowPrint [31]** automatically finds temporal correlations among destination-related features of network traffic and uses these correlations to generate app fingerprints.
- **FS-Net [11]** uses flow length sequences to classify encrypted traffic via the Gated Recurrent Unit.
- **Whisper [23, 25]** expresses traffic as frequency domain information through the fast Fourier transform and then performs robust identification.
- **ET-BERT [32]** handles the raw packets in hexadecimal and deploys a pre-trained transformer to represent and learn the contextualized datagram-level information.
- **FlowLens [13]** calculates statistical histograms of packet size distribution and adopts machine learning models (*e.g.,* XGBoost) to perform classification.
- **HyperVision [33]** is an unsupervised malicious traffic detection system that could capture flow interaction patterns represented by the graph's structural features.
- **nPrint [7]** is a tool that generates a unified packet representation and then leverages AutoML to fit the tabular data.

B.3 Distribution of Similar Intra-flow

To develop experiments, we first compute the similarity based on intra-flow features, including packet time interval, packet size, IP TTL, six TCP Flags (SYN, FIN, ACK, PSH, RST, URG), TCP window size, and UDP length. Between pairs of flows, where the similarity is a float number between 0 and 1. The cumulative distribution curves of the similarity calculation results of four groups of datasets are shown in Figure 14. We divide the interval $[0, 1]$ evenly into ten parts (*i.e.,* $\{0.1, 0.2, \cdots\ 1.0\}$), and then calculate model performance according to test data of different similarity intervals. We calculate the similarity of intra-flow features on CIC-IDS2017/2018, CrossNet2021, ISCXVPN2016, and CIC-InvesAndMal2019 datasets. Figure 14 shows that the similarity of the intra-flow features within the dataset is mainly between 0.15 and 0.4.