IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 23 August 2024, accepted 16 September 2024, date of publication 27 September 2024,
date of current version 10 October 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3469730

RESEARCH ARTICLE

Early Traffic Classification With Encrypted ClientHello: A Multi-Country Study

DANIL SHAMSIMUKHAMETOV, ANTON KURAPOV, (Member, IEEE), MIKHAIL LIUBOGOSHCHEV, AND EVGENY KHOROV, (Senior Member, IEEE)
Wireless Networks Laboratory, Institute for Information Transmission Problems of the Russian Academy of Sciences (IITP RAS), 127051 Moscow, Russia

Corresponding author: Evgeny Khorov (khorov@wireless.iitp.ru)

This work was supported by the Institute for Information Transmission Problems of the Russian Academy of Sciences (IITP RAS) funded by the Ministry of Science and Higher Education of the Russian Federation, under Project FFNU-2022-0036.

ABSTRACT Quality of service provisioning in modern networks requires traffic to be classified as quickly as possible according to its requirements and service type. However, traffic classification (TC) becomes increasingly challenging as traffic encryption evolves. The Encrypted ClientHello (ECH) amendment to the most widespread encryption protocol, Transport Layer Security (TLS), conceals the most sensitive metadata of the TLS-encrypted flows including the Server Name Indication (SNI), which provides ground-truth early TC. Nevertheless, the backward compatibility and protocol limitations leave some non-random TLS metadata open. This paper designs a new early traffic classifier called hybrid Random Forest Traffic Classifier (hRFTC) that utilizes unencrypted TLS metadata together with the statistical features of the traffic flows extracted before the arrival of any application data from the server side. The paper collects an up-to-date diversified traffic dataset in various countries of North America, Europe, and Asia, which is available online and is one of the largest, most detailed, and diversified open-source TC datasets. The paper evaluates the performance of the state-of-the-art TC algorithms on the collected dataset. The results reveal that unencrypted in ECH scenario TLS settings are similar for many multimedia services. Consequently, the TC algorithms that rely solely on the TLS features achieve as low as 38.4% classification F-score. Meanwhile, the hybrid approach of the hRFTC dramatically enhances the TC efficacy. hRFTC achieves up to a 94.6% F-score on the collected dataset, which is superior to the best state-of-the-art algorithms.

INDEX TERMS Early traffic classification, TLS, encrypted ClientHello, QUIC, random forest, machine learning, quality of service.

I. INTRODUCTION

Many network management tasks for heterogeneous Quality of Service (QoS) provisioning, such as routing [1], [2], traffic engineering [3], scheduling, and resource allocation [4], [5], require real-time identification of traffic QoS requirements [6], [7]. Traffic classification (TC) can be achieved by explicit control messaging between the application providers and the network equipment [8], [9], [10] or by passive traffic analysis by the middleboxes [11]. Both approaches have their challenges. The former requires implementing additional services and establishing contracts between content and network providers [4]. The latter, on the other hand, is not 100% reliable, because the network providers observe mostly encrypted traffic. In particular, as of 2024, above 97% of the Internet traffic is encrypted with the Transport Layer Security (TLS) protocol [12]. Hence, sophisticated algorithms are required to classify the packets by the types of their payload and set the QoS requirements accordingly.

Passive traffic analysis for QoS provisioning is often termed *early traffic classification* (eTC), because it requires recognizing the type and the QoS requirements of the packet flows as quickly as possible, ideally before the arrival of any packet with the application payload [13], [14]. It is performed by the specialized middleboxes belonging to network operators and usually proceeds as follows [15], [16].

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

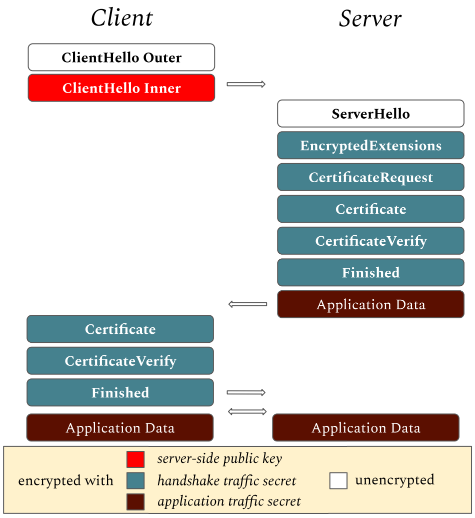

FIGURE 1. TLS 1.3 handshake with encrypted ClientHello.

First, the network operator defines a list of traffic classes and their requirements or quotas for different service providers and their traffic types. Second, the eTC algorithms observe the network packets online and assign them to one of the predefined classes. Finally, the network management tasks can be performed according to the classification results [17].

A typical TC object is a *flow*, which is a bidirectional sequence of packets sharing the same 5-tuple: IP addresses, transport layer ports, and transport protocol. Note that more granular (i.e., per-packet) TC is rarely practical. First, on the modern Internet, application providers typically store different media types on separate servers. Therefore, the clients rarely can download different data types in one connection. Second, per-packet TC is too complex because of the limited data available and too stringent classification delay requirements.

The unique characteristics associated with a specific flow are called its *features*. Researchers often differentiate between *packet-based* and *flow-based* features [13], [18]. The packet-based features include the content of packets: network and transport layer headers and the application layer payload. In contrast, the flow-based features include sequential and statistical properties of the flow, such as packet sizes and packet inter-arrival times.

Each TLS flow starts with the *TLS handshake*, which is the exchange of control data by the TLS endpoints [19]. It is required to negotiate cryptographic parameters, authenticate the endpoints, and secure the following communication. However, some of the control data during the handshake are transmitted unencrypted, which, unfortunately, discloses privacy-sensitive information and can be used by eTC algorithms. The latest version of TLS, namely TLS 1.3, encrypts all handshake messages after *ClientHello* (CH) and *ServerHello* (SH) with the *handshake traffic secret*. The unencrypted CH contains the *server name indication (SNI)* extension, which reveals the domain name of the server to any middlebox. Hence, SNI enables accurate eTC, because the servers with different media types have well-distinguishable SNIs, e.g., YouTube videos reside on servers named "r*-*googlevideo*", while YouTube Live servers are named "*rtmps.youtube" [20].

Encrypted ClientHello (ECH) protocol [21], which is currently in the final stages of standardization, enhances the security of TLS 1.3 by transmitting certain sensitive CH metadata, such as SNI, in an encrypted ClientHello Inner message. The encryption is performed with the *server-side public keys* (see Fig. 1) akin to the mechanism employed by Domain Name System over HyperText Transfer Protocol Secure (DNS-over-HTTPS) to protect DNS queries from unauthorized access. Consequently, this prevents traffic classification by SNI and makes the eTC of ECH traffic a challenging task.

However, even with ECH, some TLS data remains available for analysis by eTC algorithms. Specifically, for backward compatibility, TLS parameters that define the version of the TLS protocol, supported cipher suites, and the total length of extensions cannot be encrypted. Apart from that, the *key share*, *pre-shared key*, and *supported versions* extensions shall be unencrypted to allow the client and the server to establish shared secrets. Moreover, TLS implementations often contain so-called GREASE [22] extensions not registered by Internet Assigned Numbers Authority (IANA) [23]. According to the TLS protocol specifications, communication parties should ignore unknown extensions so that new capabilities can be introduced to the ecosystem while maintaining interoperability. Such extensions are also transmitted unencrypted.

Previous studies show that the combination of the CH and SH unencrypted metadata can still be enough for eTC in specific scenarios because traffic types often differ in TLS settings such as preferred Cipher Suite, Key Share Group, and the order of extensions [24], [25]. However, those studies have left some important questions open.

The first question is related to the generalization of the conclusions: whether the TLS-based eTC will remain accurate when the variety of source servers, traffic types, and contexts grows. Whatever eTC algorithm is used, the dataset's diversity and the accuracy of its labeling significantly influence the classification quality. Hence, to obtain robust and reliable results, we shall evaluate TC on a dataset accurately resembling modern Internet traffic. However, current publicly available datasets have several shortcomings. They are often outdated [26], [27], [28], cover few services [28], [29], or preserve few traffic details [30], [31] or lack HTTP/3 protocol traffic [24], [28]. The second open question relates to the potential value of the flow-based features for the eTC of ECH traffic because the existing flow-based TC algorithms have not yet been tested on the ECH traffic. In this paper, we address these open questions and **our main contributions are as follows.**

- We collect a dataset for the eTC that contains more than 600,000 TLS flows divided into 19 accurately labeled *target* traffic type classes. The dataset is diverse in terms of protocols, time, geography, and devices generating TLS flows. To the best of our knowledge, none of the public datasets has these properties altogether.
- We develop a novel encrypted traffic classification algorithm called *hybrid Random Forest Traffic Classifier (hRFTC)*. It uses the unencrypted payload of the TLS handshake, flow-based time series, and packet size statistics as independent classification features. We show that hRFTC outperforms the state-of-the-art classifiers, namely, packet-based [24], [32], [33], flow-based [30], and hybrid [34], [35], in classification quality.
- We study how well the state-of-the-art eTC algorithms can handle the heterogeneity of the data in the ECH scenario and how well they generalize from a small amount of it.
- We demonstrate that even the best hybrid TC algorithms trained in one geographic location shall be retrained in another due to notable differences in traffic patterns.

We organize the rest of this paper as follows. Table 1 lists the abbreviations used throughout the paper. Section II observes the recent studies that proposed TC algorithms that can classify ECH traffic. Then, we develop the novel hybrid TC algorithm in Section III. In Section IV, we describe the collected heterogeneous traffic dataset. Then, Section V describes the metrics we use to evaluate the performance of the TC algorithms and the similarity of the considered traffic classes. Finally, in Section VI, we compare the developed algorithm with other baseline TC algorithms on the collected dataset, and we conclude the paper with Section VII.

TABLE 1. List of used abbreviations.

Abbreviation	Meaning
AB-RF	Aligned-Bytes Random Forest (TC algorithm)
BA	Buffered Audio
BGRUA	Bidirectional Gated Recurrent Unit with Self-Attention (TC algorithm)
BV	Buffered Video
CDN	Content Delivery Network
CH	ClientHello (TLS message)
CNN	Convolutional Neural Networks (NN algorithm)
CPU	Central Processing Unit
DL	Downlink
DNS	Domain Name System (L7 protocol)
ECH	Encrypted ClientHello (L6 protocol)
eTC	Early Traffic Classification
GI	Gini-Impurity
HTTP	HyperText Transfer Protocol (L7 protocol)
HTTPS	HyperText Transfer Protocol Secure (L7 protocol)
LSTM	Long Short-Term Memory (NN algorithm)
LV	Live Video
IPT	Inter-Packet Time
MATEC	Multi-head Attention Encoder (TC algorithm)
ML	Machine Learning
MLP	Multi-Layer Perceptron (NN algorithm)
NN	Neural Network
PS	Packet Size
QoS	Quality of Service
QUIC	Quick UDP Internet Connections (L4 protocol)
RB-RF	Recomposed-Bytes Random Forest (TC algorithm)
RF	Random Forest (ML algorithm)
hRFTC	Hybrid Random Forest Traffic Classifier (TC algorithm)
SBV	Short Buffered Video
SH	ServerHello (TLS message)
SNI	Server Name Indication (TLS extension)
SVM	Support Vector Machine (ML algorithm)
TC	Traffic Classification
TCP	Transmission Control Protocol (L4 protocol)
TLS	Transport Layer Security (L6 protocol)
UDP	User Datagram Protocol (L4 protocol)
UL	Uplink
UW	University of Waterloo Tripartite algorithm

II. EARLY TRAFFIC CLASSIFICATION BACKGROUND

Let us divide the existing eTC algorithms into *packet-based*, *flow-based*, and *hybrid* according to the features used for classification. Here, *hybrid* algorithms combine packet-based and flow-based features. The following sections review the relevant algorithms of each of these classes, and Table 2 summarizes the best algorithms capable of eTC.

A. PACKET-BASED TC ALGORITHMS

All packet-based algorithms for encrypted TC are inherently *eTC* algorithms because the TLS handshake remains the only informative part of the flow, with encryption completely masking the payload of every subsequent application data packet. We can categorize packet-based algorithms by their feature extraction method.

The most popular approach is based on neural networks (NN). NN-based algorithms train to extract features within their hidden layers from the raw bytes of the first packets of the encrypted flow. The state-of-the-art examples include Stacked Auto-Encoder [36], 1-/2-dimensional Convolutional Neural Networks (1D/2D-CNN) [37], [38], CNN combined with Long Short-Term Memory (CNN-LSTM) [39], Capsule Neural Network [40], Bidirectional LSTM with Attention Mechanism [41], Bidirectional Gated Recurrent Unit with Self-Attention Mechanism (BGRUA) [32], Multi-head Attention Encoder (MATEC) [33], Multi-Task Transformer [42], light-weighted Packet Feature N-gram Embedding with Multi-Layer Perceptron (MLP) [43], and a neural architecture search with a Recurrent Neural Network controller to generate NN architectures called AutoML [25]. However, none of these studies has implemented the ECH scenario, raising concerns about potential biases in their results. Namely, they utilized SNI as a label while masking it to hide from the NN. However, that left some information that could indirectly reveal SNI, such as SNI length or padding length [24].

Another approach involves predefined feature extraction algorithms that use a fixed-size feature vector as an input to a machine learning (ML) classifier. The state-of-the-art examples include two Random Forest (RF)-based algorithms [24], Aligned-Bytes Random Forest (AB-RF) and Recomposed-Bytes RF (RB-RF). It has been demonstrated that RB-RF and AB-RF outperform NN-based algorithms, specifically MATEC and BGRUA, in classification quality and complexity, achieving a rather low 7.3% error rate even with ECH. The paper introduced a framework to simulate ECH from TLS 1.2 and TLS 1.3 traffic traces. However, the study was conducted on a relatively small traffic dataset with 12 classes and 3547 flows. TLS evolution and increasing traffic variety can disable this type of TC by limiting the distinguishable data.

In summary, the main advantage of NN-based classifiers is in eliminating the need for expert-driven feature extraction algorithms. However, they require substantial computational resources for training and inference, challenging their real-time operation on less powerful devices. Despite the growing trend towards developing less resource-intensive NN solutions [25], [33], [43], they still lag behind traditional machine learning methods in speed [24]. Moreover, while analyzing raw data, NNs might misinterpret encrypted traffic noise as meaningful patterns. It leads to poorer performance compared to methods based on predefined features. Lastly, NN-based classifiers lack the interpretability of approaches like RB-RF.

B. FLOW-BASED TC ALGORITHMS

Flow-based TC algorithms derive features from the sequences of packet sizes (PSs) and inter-packet times (IPTs) [18], [44], [45]. Such algorithms do not rely on unencrypted data, making them suitable even for fully encrypted traffic classification, such as TC of Virtual Private Network (VPN) traffic [46], [47]. Flow-based algorithms could also be categorized by feature extraction approach into classic ML-based and NN-based ones.

Classic ML algorithms for TC are based on accurate feature engineering. The authors of [48] highlight the significance of IPTs for TC by protocol. Following this, [49] employed a Support Vector Machine (SVM) classifier, leveraging statistical measures, such as the 25th and the 75th percentiles, median, minimum, maximum, and average from sequences of PSs and IPTs in the downlink, uplink, and mixed directions. The study achieved varying levels of accuracy, reaching up to 90% with the analysis of 3000 packets of each flow. However, the delay of collecting 3000 packets per flow is too high for eTC. Further, [50] extended the feature set with the number of packets per direction, total bytes per direction, the variance of PS and IPT sequences, and the PS distribution categorized by number of packets with packet size belonging to one of the following intervals (0, 64], (64, 128], (128, 256], or (256, 512] bytes. However, the approach includes observation of flows for 5 minutes to gather features before classification, which also contradicts the eTC requirements. The same issues have the more recent papers [51], [52], and [53].

Flow-based TC algorithms with NNs mostly focus on exploring new methodologies rather than introducing new features. The authors of [54] showed that combining RF with a Generative Adversarial Network (GAN) could slightly improve RF's performance. Next, [55] proposed converting PS and IPT sequences into an image format for classification using a 2D-CNN, which was found to outperform traditional classifiers such as SVM, Decision Tree, and MLP. This method achieved high protocol and service TC accuracy by the first 10 packets of the flow. However, considering the server IP address as a feature raises concerns about the potential biases reflecting the dataset's collection methodology. Concurrently, [56] concentrated on utilizing flow-based features to classify various protocols, such as HyperText Transfer Protocol (HTTP), Secure Sockets Layer (SSL), DNS, Quick UDP Internet Connections (QUIC), and Simple Mail Transfer Protocol (SMTP). The proposed algorithm employs a CNN+LSTM network architecture that examines the first 20 packets. This approach resulted in an 84% accuracy rate using solely flow-based features, which rose to 96% when the model was enhanced with additional features, such as source and destination ports. However, these ports can indicate specific protocols, for example, port 443 is used for HTTPS and port 80 for HTTP, which is not useful in the QoS classification scenario we consider in this paper.

Paper [57] considered the problem of in-application activity identification, e.g., sending a message, sending a friend request, or starting a voice call. For classification, the developed algorithm considers all packets within an up to 0.5 seconds-long time-window, which is a viable time-window for eTC. Similarly, paper [58] enhanced the traffic classification accuracy by dividing the classification process into two phases. For some flows, the high classification accuracy is achieved with only TLS handshake packets, for others, around 20 packets are considered. This way, the authors try to strike a balance between classification accuracy and delay.

Next, a few papers, e.g., [59] and [60], explored the use of Graph representation for flow-based features and employed an MLP Graph NN for the TC problem, which also requires collecting a large number of packets per flow and does not fit the eTC criteria. Paper [61] presented a novel NN architecture to analyze PSs and IPTs, reporting perfect classification quality on 10 to 50 packets. Unfortunately, the paper considered an outdated dataset as well as [62]. Finally, recently, paper [63] extended the service identification problem by allowing the algorithm to classify some services as unknown if the flow does not fit into any predefined classes. The authors proposed the CESNET model with a multi-modal architecture, where the CNN processes the time-series flow representation of PSs, IPTs, and directions,

TABLE 2. Summary of most notable early traffic classification studies.

Feature Type	Ref.	NN/ML	Study Year	Classification Problem	Traffic	ECH	Dataset Size, Number of flows	Dataset Year
Packet-based	[38]	NN	2017	Traffic Type	Multi-protocol	No	160k	2016
	[39]	NN	2018	Traffic Type	Multi-protocol	No	260k	2016
	[40]	NN	2019	Traffic Type	Multi-protocol	No	260k	2016
	[41]	NN	2019	Traffic Type	Multi-protocol	No	260k	2016
	BGRUA, [32]	NN	2020	Service	TLS (hidden SNI)	No	590k	2016
	MATEC, [33]	NN	2021	Service	TLS (hidden SNI)	No	590k	2016
	[42]	NN	2022	Traffic Type	Multi-protocol	No	260k	2016
	RB-RF, [24]	ML	2022	Service & Traffic Type	TLS+ECH	Yes	3.5k	2021
	[25]	NN	2023	Service & Traffic Type	TLS (hidden SNI)	No	380k	2021
Flow-based	[55]	NN	2017	Protocol & Service	Multi-protocol	No	22k	2017
	[56]	NN	2017	Protocol	Multi-protocol	No	260k	2017
	[46]	ML	2020	Traffic Type	Multi-protocol	No	260k	2016
	[57]	NN	2022	Service	TLS	No	65k	2022
	CESNET, [63]	NN	2023	Service & Traffic Type	TLS	No	140M	2022
Hybrid	**hC4.5**, [34]	ML	2020	Service	TLS (hidden SNI)	No	590k	2016
	[64]	NN	2022	Service	TLS (hidden SNI)	No	240k	2018
	UW, [35]	NN	2023	Service & Traffic Type	TLS (hidden SNI)	No	450k	2021

Note: We emphasize with **a bold font** the algorithms considered in this paper as baselines.

while linear layers handle pre-extracted statistical flow features in parallel. CESNET achieved 97% accuracy when considering the first 30 packets of the flow on a vast up-to-date dataset.

To conclude, first, many of the flow-based TC algorithms consider long sequences of packets available for TC, therefore, they fail to meet the eTC requirements. Second, many studies incorporated traffic collection artifacts from Transmission Control Protocol (TCP) and Internet Protocol (IP) packet headers, potentially skewing the results. Third, none of the reviewed studies considered the ECH TC problem. Meanwhile, [24] shows that the CH length can reveal the SNI and the associated server. Therefore, without proper adjustments to the CH length, the flow-based TC algorithms can classify traffic by SNI, which is a trivial and long-known solution [20].

C. HYBRID TC ALGORITHMS

Finally, hybrid TC algorithms combine the benefits of both packet-based and flow-based ones. For example, [34] shows that TLS handshake features combined with statistics over ten application packets are enough for accurate eTC with a decision tree-based C4.5 algorithm. However, the authors used a small subset of payload and flow features, which can hinder the algorithm from showing the best performance on more complex data. More recently, paper [64] considered the TLS handshake and flow statistics for the first ten packets of the flow and reported a rather high classification accuracy. However, like many authors of the papers on flow-based TC, the authors simply mask SNI in a CH message, which is not enough to protect the flow from being classified by SNI [24].

Paper [65], in turn, applies hybrid TC algorithms to detect encrypted malware traffic. The authors significantly improved the classification quality by extending TLS handshake features with flow statistics. Unfortunately, the developed algorithm analyzes the first 50 packets of each network flow, which incurs a high classification delay. So, it is incapable of eTC, as well as the algorithms described in paper [66], which considers the entire packet flows.

Recently, the authors of the paper [35] proposed the University of Waterloo Tripartite algorithm (UW), which incorporates a multi-modal architecture. This architecture includes a CNN to extract raw TLS handshake bytes, an LSTM to process time-series flow representations, including a three-dimensional array of PSs, IPTs, and directions, and an MLP to handle statistical flow characteristics. The UW algorithm demonstrates superior performance compared to baseline algorithms in classifying TLS traffic by type and service provider. However, the study does not consider ECH encryption or the eTC scenario.

To sum up, the existing eTC algorithms either have low accuracy and high classification delay or need to be reexamined on more up-to-date and diverse datasets with accurately simulated ECH.

III. HYBRID RFTC

In this section, we describe a novel eTC algorithm called hRFTC. The algorithm enhances our previous packet-based classifier, RB-RF [24], in the following ways. First, it extends the RB-RF functionality to handle QUIC traffic flows. Second, its novel packet selection criterion allows it to improve the TC quality while satisfying the eTC classification delay requirements. Third, hRFTC incorporates a novel set of flow-based features. Fig. 2 presents the overall scheme of the developed algorithm.

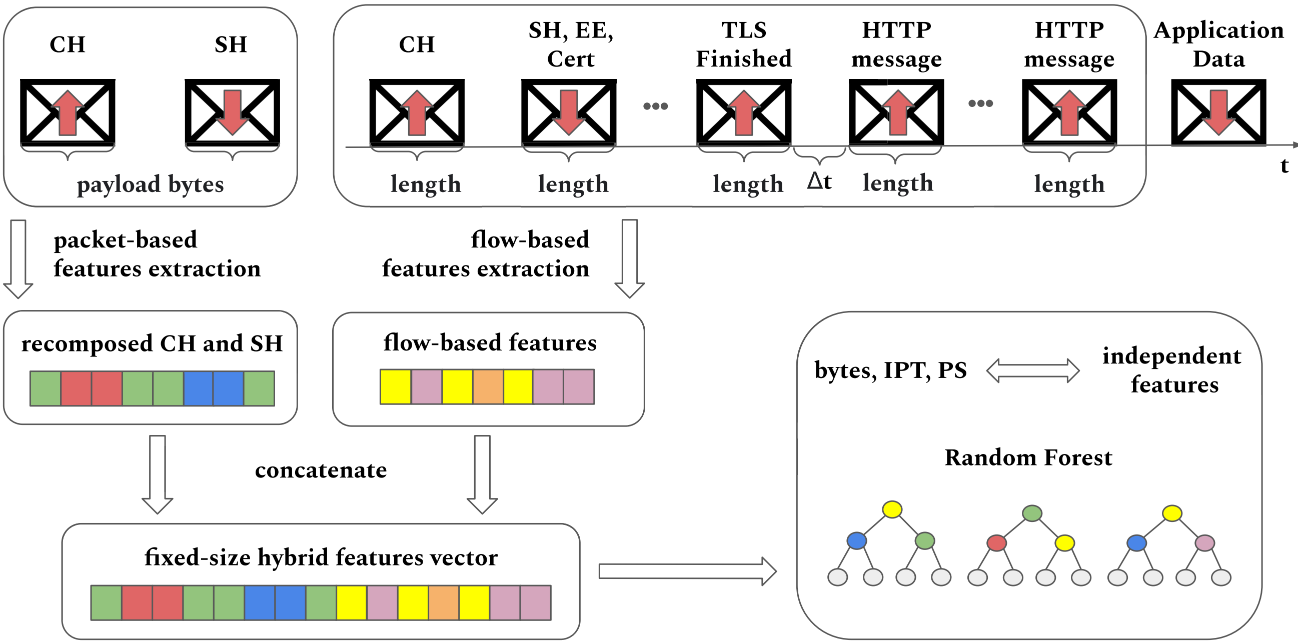

FIGURE 2. Hybrid random forest traffic classifier.

A. INTRODUCTION TO RB-RF

The main idea of the original RB-RF [24] classifier is to restructure the payload bytes of the unencrypted TLS handshake messages and treat them as independent classification features. For each unencrypted parameter, the algorithm assigns a particular position and length in a predetermined fixed-size byte vector, that we call *recomposed payload*. This vector is then fed into the Random Forest (RF) classifier.

Although RB-RF can extract packet-based features from both TCP/IP headers and payload, it focuses on *payload-based* features for several reasons. First, IP headers often carry data collection artifacts that can bias the TC results [63]. For example, specific data collection locations can influence IP addresses. Furthermore, network operators' frequent adjustments to the Differentiated Services Code Point value can skew the traffic dataset based on the collection point, leading to TC inaccuracies [17], [67]. Second, most fields of TCP and UDP headers, such as the sequence number, which is chosen at random at the start of the TCP three-way handshake, or a destination port, e.g., 443, which is the standard for HTTPS flows, do not provide information valuable for TC.

B. RB-RF EXTENSION TO QUIC

The latest version of the HTTP protocol, HTTP/3, replaces the combination of TCP and TLS lower-layer protocols with the combination of UDP and QUIC respectively. Given that HTTP/3 already generates around 30% of global HTTP traffic [68], TC algorithms shall be able to classify HTTP/3 (QUIC) traffic. Meanwhile, RB-RF was initially designed to process TLS-over-TCP packets, so we extend it in this paper to consider the following peculiarities of QUIC protocol.

First, unlike TCP, where each connection owns two sockets at the endpoint, multiple QUIC connections can share the same UDP socket and a pair of ports. Consequently, for a traffic classification middlebox, multiple QUIC connections can appear as a single flow, i.e. a group of packets united by the 5-tuple: IP addresses, transport layer ports, and transport protocol. However, to demultiplex packets belonging to different QUIC connections QUIC uses connection IDs, which are specified in QUIC packet headers. Thus, for QUIC, we consider a flow to be the sequence of packets sharing the same QUIC connection ID.

Second, QUIC endpoints encrypt CH and SH messages with the keys derived from the Destination Connection ID of the first Initial Packet and a specific salt that is defined in the specification of each QUIC version [69]. Such an approach protects QUIC traffic from spoofing by the middleboxes unaware of the version and its salt for the observed QUIC flow. However, it cannot protect from a version-aware middlebox [70]. Thus, in our experiments, we assume that the TC middlebox can keep a database of the existing QUIC versions and can decrypt and analyze CH and SH messages.

Third, Initial packets of a QUIC connection can contain CH or SH messages encapsulated in so-called CRYPTO frames together with other service messages encapsulated in PING, PADDING, or ACK frames [69]. However, as these frames do not belong to the TLS protocol, we do not consider deriving payload features from them. Moreover, paper [71] shows that QUIC PADDING frames cannot protect from traffic analysis.

Finally, the structure of CH and SH messages in TLS-over-TCP differs from the structure of these messages in QUIC. Specifically, the TLS Record header in QUIC lacks

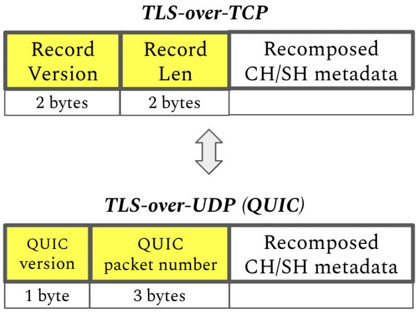

FIGURE 3. Recomposed payload modification for QUIC flows.

TABLE 3. Selected flow-based feature set in hRFTC.

Category	Description
*N of Packets**	Count of DL and UL packets with non-zero L4 payload
↑↓ *PS Stats*	Mean, standard deviation, min, max, 25th, 50th, 75th percentile, sum of PSs
↑↓ *PS Pattern*	Aligned vector of the first six PSs
↑↓ *PS Unique*	Aligned vector of six largest unique PSs
↑↓ *PS Histogram*	Vector of PS frequencies for bins of 250 bytes
↑↓ *IPT Stats*	Mean, standard deviation, min, max, 25th, 50th, 75th percentile, sum of IPTs

* *until the first downlink packet with application data.*

the Version and Length fields analyzed in the original RB-RF. In contrast, QUIC packets introduce additional fields, such as QUIC Version and QUIC packet number, which are used in generating a cryptographic nonce. The latter field can differ from the actual sequence number of the packet in a flow and this way it can indirectly identify the endpoints. Consequently, we substitute the missing fields of the TLS Record layer with the relevant QUIC-specific fields in the recomposed payload as shown in Fig. 3.

C. FLOW-BASED FEATURES

We derive flow-based features from the sequences of PSs and IPTs. In eTC, the flow-based features introduce a tradeoff between classification accuracy and delay. Specifically, the increasing number of packets analyzed by a TC algorithm improves the TC accuracy. However, at the same time, it increases the TC delay and degrades the efficiency of the following network management procedures.

Unlike the previous studies, we address this tradeoff by taking into account the *the actual applications workflows*. Instead of considering a predefined number of analyzed packets [34], [65], we consider *all packets until the first downlink packet with application data.* The obtained packet sequence includes all TLS handshake packets and the initial client requests. The considered few first service packets carry no application data. Hence, from the application perspective, the TC delay caused by the collection of these packets insignificantly slows down the QoS management.

For the collected sequence, we consider only the packets with non-zero transport-layer payload and derive a set of features defined in Table 3 separately for downlink (DL) and uplink (UL) packets. Although the PSs and IPTs are known to exhibit highly non-linear properties [72], we avoid exploring the high-order moments, or self-similarity of their time series. To achieve low classification delay, our algorithm operates with statistics over approximately a dozen packets, which is too few to estimate the complex statistical measures accurately.

For each traffic flow, we combine flow-based features into one feature vector and concatenate it with the recomposed payload. The obtained hybrid feature vector serves as an input of the Random Forest algorithm (see Fig. 2).

IV. THE COLLECTED TLS DATASET

In this section, we present the collected multi-country TLS traffic dataset. Then, we describe how we simulate ECH from the existing TLS 1.2/1.3 flows.

A. THE DATASET DESCRIPTION

To evaluate eTC algorithms in realistic conditions, we carefully collected and accurately labeled a heterogeneous multi-country TLS traffic dataset. Using Wireshark [73], we captured traffic in separate traces and named them by the service and QoS class this traffic belongs to. We collected the traces between October 2022 and January 2023 in various locations, including Germany (Munich), Kazakhstan (Aktau, Aktobe), Russia (Dolgoprudny, Moscow, Zelenograd), Spain (Girona), Turkey (Antalya, Kemer), and the USA (Miami). Table 4 shows the distribution of collected traffic volume over the locations.

TABLE 4. Traffic volume distribution over geographic locations.

Country	Cities	Number of Flows
Germany	Munich	121,936
Kazakhstan	Aktau, Aktobe	17,228
Russia	Moscow, Dolgoprudny, Zelenograd	234,335
Spain	Girona	95,154
Turkey	Antalya, Kemer	160,131
USA	Miami	43,850

When choosing the set of collected traffic types, we aimed to reproduce the structure of the traffic on the modern Internet but also to collect the traffic with sufficiently different QoS requirements. Namely, as of today, video traffic remains the most prevalent traffic type. According to Sandvine's report [74], it contributes to more than 67% of worldwide Internet traffic. Among video traffic, the *short buffered video* (SBV) traffic has the highest growth rate. It includes the traffic generated by Instagram Reels, YouTube Shorts, TikTok, and other similar services. However, the *long buffered video* (BV) has the largest overall volume. Moreover, its main contributors, YouTube and Netflix, generate more than a quarter of the world's total traffic volume. In addition to SBV and BV traffic, the dataset includes *buffered audio* (BA) (generated by Spotify, Apple Music, etc.), and upstream *live video* streaming (LV) as other huge contributors to the worldwide Internet traffic. Finally, the dataset contains TLS-over-TCP and QUIC-over-UDP web traffic as a heterogeneous background traffic class.

We collected traffic using Safari, Google Chrome, and Firefox web browsers on personal computers running macOS, Windows, and Ubuntu operating systems. Additionally, we used OBS Studio for upstream LV streaming to platforms such as YouTube and Facebook. For mobile devices, we captured BV and BA traffic using the latest versions of the corresponding applications on iOS and Android smartphones. Lastly, we gathered web page download traces of 3500 most popular web pages, according to [75].

TABLE 5. The multi-country TLS dataset classes and SNI labeling patterns.

No	Traffic Type	Service	Num. of Flows	SNI Pattern
1	buffered audio	AppleMusic	1015	*od*itunes.apple.* *audio*itunes.apple.*
2		SoundCloud	2255	*cf*hls*media*sndcdn*
3		Spotify	1054	*audio*spotify*akamai* *audio*scdn*
4		VkMusic	1337	*vkuseraudio*
5		YandexMusic	1487	*storage*yandex.*
6	short buffered video	Instagram Reels	1994	*instagram*fbcdn*
7		TikTok	1401	*tiktokcdn*
8		VkClips	1290	*vkvd*
9		YouTubeShorts	3022	r*-*googlevideo.*
10	buffered video	Amazon PrimeVideo	2559	*row.aiv-cdn* *avod*akamai*
11		Facebook Video	4472	*video*fbcdn.net scontent*.fbcdn.net
12		Kinopoisk	1257	*strm*yandex.*
13		Netflix	3594	*nflxvideo*
14		Vimeo	1329	*vod-adaptive*akamai*
15		VkVideo	1441	*vkvd*
16		YouTube	1246	r*-*googlevideo*
17	live video	Facebook Live	1041	*rtmp-api.facebook.*
18		YouTube Live	1194	*rtmps.youtube* *upload.youtube*
19	other/web	Any	639646	others

We used Selenium Webdriver [76] to automate traffic collection on personal computers for most of the collected data. However, some traffic, namely, all non-web traffic from Kazakhstan and Turkey, was collected manually. To facilitate straightforward labeling, we isolated each type of traffic during the collection phase. This approach was necessary because various traffic types from the same multimedia provider might share identical SNI patterns. For instance, both services, YouTube videos and YouTube Shorts, have the same SNI pattern: *r*-*googlevideo**.

According to [74], the share of QUIC traffic on the modern Internet is about 10%. Namely, YouTube generates 46% of all world QUIC traffic, while Instagram Reels and Facebook Videos generate 17% and 5% of QUIC traffic, respectively.

TABLE 6. Summary of the multi-country TLS dataset.

Parameter	Details
Continents	North America, Europe, Asia
Countries	USA, Spain, Kazakhstan, Turkey, Russia, Germany
Cities	Miami, Girona, Aktau, Aktobe, Munich, Kemer, Antalya, Moscow, Dolgoprudny, Zelenograd
Operating Systems	MacOS, Windows, Ubuntu, iOS, Android
Devices	8 PCs, 5 Smartphones
Traffic Types	buffered audio, short buffered video, buffered video, live video, web
Multimedia Providers	Apple Music, SoundCloud, Spotify, Vk Music, Yandex , Instagram, TikTok. YouTube, Netflix, Amazon Prime, Kinopoisk, Vimeo, Facebook
Web Pages	3,500 most popular [75]
TLS-Encrypted Flows	>600,000
Protocol Types	TLS-over-TCP, QUIC
Collection Period	4 months (2022-2023)

In our dataset, QUIC traffic is represented by YouTube videos, Facebook Videos, Instagram Reels, their QUIC web traffic, and web page download traces of 350 QUIC web pages from [77]. Overall, the dataset includes both TLS-over-TCP flows (90.7 %) and TLS-over-UDP QUIC flows (9.3 %). Table 5 shows the number of collected flows for each traffic type and service.

As ECH is under development and is not widely adopted yet [78], the CH messages in the collected traffic contain plaintext SNIs. So, we use the SNI patterns listed in Table 5 to label each pair of service and traffic type as a *target* class, while the web traffic is considered the *background* class. If different traffic types generated by the same service share a common SNI pattern (e.g., in the case of YouTube and YouTube Shorts, VkClips, and VkVideo), we label them based on the name of the traffic category saved in the capture file name during the collection. We treat any non-multimedia flows generated by the multimedia services (e.g., a web page containing a YouTube video player) as web because the QoS requirements for this traffic are closer to a web rather than to a multimedia stream.

Overall, the dataset consists of 18 target classes, each with over 1,000 flows, along with a massive background class encompassing 639,646 flows. The dataset is available at https://wireless.iitp.ru/qos-tls-dataset-2023/. Table 6 summarizes the dataset diversity.

B. ECH SIMULATION

As the TLS Encrypted ClientHello protocol is still under development [21], and ECH traffic is not widely spread, we modify the collected dataset to simulate ECH. Specifically, we preprocess the collected traffic before classification to simulate the ECH-encrypted CH and SH messages. We consider the strongest ECH protection against TC and leave open only those TLS extensions that cannot be encrypted with ECH: key share, pre-shared keys, and TLS versions.

Unfortunately, the current ECH version has not yet been extended to QUIC. Hence, we assume the following ECH implementation for QUIC. First, all fields and extensions common to QUIC and TLS can be protected similarly. Second, QUIC integrates the transport layer handshake into the TLS handshake by defining a TLS extension named QUIC Transport Parameters. This TLS extension contains a set of transport parameters registered in [70], which can have various lengths and positions inside the TLS extension. According to [79], the order of transport parameters helps to distinguish some QUIC implementations. However, other implementations randomize these positions. So, to consistently consider the worst-case ECH modification, we exclude the QUIC Transport Parameters extension from our analysis.

V. METRICS

A. CLASSIFICATION QUALITY METRICS

To evaluate the efficiency of hRFTC, we use a widely-known F-score metric [80], calculated as follows. Let *precision* be the portion of correctly classified class members among all items attributed to this class. Let *recall* be the portion of correctly classified members of the class among all the members of the class. Then, *F-score* for the class is the harmonic mean of *precision* and *recall* for that class. Finally, *macro F-score* is the arithmetic mean over all the per-class *F-scores*. We choose the macro F-score because it considers the F-score of each class and then takes the average, treating all classes equally regardless of their cardinality. Unlike accuracy or weighted F-score, the macro F-score ensures that the algorithm performance for the minority classes is not overshadowed by the majority classes [80].

B. TLS DISTINCTION METRIC

As the quality of packet-based TC depends on the distinction between the TLS parameters (e.g., Cipher Suite, Key Share Group) of different traffic types, we use a *TLS distinction metric* first described in [81]. This metric measures the similarity of client and server parameters indicated in CH and SH messages for different target classes. Thus, we can estimate how much information is available to differentiate these classes based on their CH and SH messages. The metric is calculated as follows.

Let us denote by the *TLS fingerprint* of the packet flow the concatenated recomposed payload of CH and SH as described in Section III. Note that the recomposition takes only TLS parameters with non-random values. Thanks to that, TLS fingerprints of the handshakes generated by the same client and server at different time instants will likely be the same.

Then, let the dataset contain flows with F different TLS fingerprints belonging to C traffic classes. Let N_{fc} be the number of TLS handshakes from class $c \in [1, C]$ having fingerprint $f \in [1, F]$. Then the *TLS distinction* metric d_{fc} for the fingerprint f in class c is the share of class c TLS handshakes with fingerprint f among all TLS handshakes with fingerprint f [81]:

$$d_{fc} = \frac{N_{fc}}{\sum_{k=1}^{C} N_{fk}}.$$

The more objects of other classes have fingerprint f, the less unique it is for class c and vice versa. Using the TLS distinction, we can define the distinction D_c of a class c as the total distinction of all its TLS handshakes in relation to the total number of TLS handshakes of this class c [81]:

$$D_c = \frac{\sum_{f=1}^{F} N_{fc} d_{fc}}{\sum_{f=1}^{F} N_{fc}}.$$

TABLE 7. Example of TLS distinction calculation on buffered audio subset.

Traffic Class	$N_{fc}(d_{fc})$			D_c	Avg Distinction
	f_1	f_2	f_3		
Class1	100 (0.29)	100 (0.19)	100 (0.67)	0.38	0.45
Class2	50 (0.14)	20 (0.04)	0 (0.00)	0.11	
Class3	200 (0.57)	0 (0.00)	50 (0.33)	0.52	
Class4	0 (0)	400 (0.77)	0 (0.00)	0.77	

Finally, the dataset *average distinction* $\mathbb{D}$ is the average distinction over its classes [81]:

$$\mathbb{D} = \frac{1}{C} \sum_{c=1}^{C} D_c.$$

Table 7 shows an example of the TLS distinction calculation for a dataset with four different classes and three different TLS fingerprints.

The dataset average distinction is at its maximum and equals to 1 if all classes have different TLS fingerprints, i.e. if $\forall f, c, f \in [1, F], c \in [1, C]$: $d_{fc} = 1$, then $D_c = 1$ and $\mathbb{D} = 1$.

Conversely, the dataset average distinction is at the minimum and equals $1/C$ if all classes have the same TLS fingerprint. More formally, if $\forall f \in [1, F]\ c \in [1, C]$: $d_{fc} = N_c/N$, where N_c is the number of TLS fingerprints of class c. Then, $D_c = N_c/N$ and $\mathbb{D} = 1/C$, where N is the total TLS fingerprint count.

The described metric helps measure the intersections between the classes by non-random TLS parameters. While it does not help us to determine the exact traffic category, it allows us to estimate how diverse the TLS fingerprints of the classes are and how frequently the same fingerprints belong to the flows of different classes.

VI. NUMERICAL RESULTS

This section evaluates the TC quality of the developed and existing algorithms on the collected TLS traffic dataset. First, we analyze the TC quality of packet-based algorithms on the collected dataset and its subsets and explain their poor results. Second, we compare the TC quality of the hRFTC algorithm and the best state-of-the-art ones on the entire dataset. Third, we estimate the importance of TLS and statistical features for TC. Finally, we study how well hRFTC can generalize from a small amount of data or traffic collected in different countries.

A. EXPERIMENT SETUP

In each experiment, unless stated otherwise, we divide the traffic dataset into training, validation, and test subsets in a 7:1:2 ratio, ensuring that each class is proportionally represented through stratified sampling. To ensure reliable and robust results, we average the outcomes over 20 independent dataset splits. For each split, we conduct 5 independent runs of classifiers, resulting in a total of 100 runs.

As baseline eTC algorithms, we consider the hC4.5 [34], UW [35] hybrid algorithms, the CESNET flow-based algorithm [63], and three packet-based eTC algorithms: BGRUA [32], MATEC [33], and RB-RF [24]. The baseline algorithms are executed with the optimal hyperparameter values according to the original studies. To ensure eTC, instead of using a predefined number of analyzed packets, we consider all packets until *the first downlink packet with application data after the TLS handshake* for hC4.5, UW, and CESNET, consistent with proposed hRFTC. Regarding hRFTC, its Random Forest classifier has the hyperparameter values shown in Table 8. We do not limit the depth of the decision trees to achieve optimal classification quality. However, this does not incur computational costs as the average depth of the trees observed in our experiments is 38, and the maximum depth is 49.

TABLE 8. Optimal hyperparameters for hRFTC.

Hyperparameter	Decision Trees	Max Features	Max Depth
Value	150	0.3	unlimited

All the results are obtained on the workstation described in Table 9. On such a workstation and with such hyperparameters, we obtain an inference rate slightly higher than 100,000 flows per second for the hRFTC algorithm, which is much higher than any results reported by recent NN-based eTC classifiers [24], [43]. Therefore, such an algorithm can be easily run even on central processing units (CPUs) installed in low-end off-the-shelf Wi-Fi access points [17].

TABLE 9. The workstation configuration.

Item	**Configuration**
CPU	AMD Ryzen 7 3700X
RAM	2x16GB @ 3200MHz
Operating System	Ubuntu 18.04.5 LTS
Python version	Python 3.6.9
Tensorflow version	Tensorflow 2.4.0-rc4
Sklearn version	Sklearn 0.22.2.post1

B. ECH: PACKET-BASED ALGORITHMS AND TLS DISTINCTION

We start by investigating the classification performance of the best-known packet-based algorithms, namely, BGRUA, MATEC, and RB-RF. We aim to assess the similarity of classes by TLS parameters in the ECH scenario. To simplify evaluation, we focus on dataset subsets that either include all services generating a specific traffic type (e.g., SBVs from Instagram, TikTok, YouTube Shorts, and VkClips) or all traffic types from services of a specific company (e.g., Google-owned services: YouTube, YouTube Shorts, YouTube Live, and Google-related web). To obtain the web class of a particular service, we extract a subset from the general web class using SNI masks corresponding to the considered service. The following SNI patterns cover all of the flows labeled as web traffic of a specific service:

TABLE 10. Comparison of the packet-based classifiers on the subsets of the dataset.

Selected Traffic Subset	**Avg TLS Distinction of the Traffic Subset**	**Macro F-score [%]**		
		RB-RF	**BGRUA**	**MATEC**
Live Video	1	100.0	100.0	100.0
Short Buffered Video	0.99	99.2	81.5	82.3
Buffered Video	0.87	87.4	75.7	75.9
Buffered Audio	0.71	71.8	59.9	63.0
Vk	0.65	66.2	57.1	61.1
Google	0.58	55.5	55.0	54.7
Facebook	0.53	52.6	51.4	51.2
Yandex	0.49	40.2	39.8	38.1

- Vk web: **vk*, sun*userapi*, *mycdn.me.*, *mail.ru* ;
- Google web: **google*, *doubleclick*, *2mdn*, *.gvt*, *youtube*, *ytimg*, *ggpht*, *gstatic** ;
- Facebook web: **facebook*, *fbcdn*, *cdn.fbsbx*, *instagram**;
- Yandex web: **yandex*, *kinopoisk**.

Table 10 confirms that as the TLS distinction decreases, the performance of all packet-based classifiers decreases reaching about 40% macro F-score for the most similarly configured services of Yandex. This indicates that packet-based classifiers struggle to differentiate traffic types with similar TLS settings in the presence of ECH. Additionally, the table highlights that different services often configure TLS similarly. For example, the majority of BA servers in our dataset prefer the same set of Cipher Suites, which are mandatory-to-implement in TLS 1.3 [19]. For instance, the *TLS_AES_128_GCM_SHA256* Cipher Suite is chosen by 95% of SoundCloud servers and 50% of AppleMusic servers. Similarly, all VkMusic and YandexMusic servers, along with 45% of AppleMusic servers, select the *TLS_AES_256_GCM_SHA384* Cipher Suite.

C. ECH: HYBRID VS. FLOW-BASED VS. PACKET-BASED CLASSIFIERS

The results of Section VI-B confirm that ECH makes packet-based classifiers inefficient for QoS-aware eTC. Consequently, pure packet-based eTC algorithms have to be updated for new versions of TLS. To assess the effectiveness of the hybrid algorithms: hRFTC which is developed in the

TABLE 11. Full dataset per class F-score for different classifiers.

Class	F-score [%]						
	Hybrid Classifiers			Flow-based Classifier	Packet-based Classifiers		
	hRFTC [proposed]	**UW** [35]	**hC4.5** [34]	**CESNET** [63]	**RB-RF** [24]	**MATEC** [33]	**BGRUA** [32]
BA-AppleMusic	**92.1**	89.5	80.2	89.2	25.5	13.1	14.5
BA-SoundCloud	**99.6**	98.9	97.8	98.7	84.4	81.8	82.0
BA-Spotify	**93.6**	90.8	89.0	88.5	16.3	0.0	3.6
BA-VkMusic	**95.7**	89.7	88.5	91.8	2.6	2.1	3.2
BA-YandexMusic	**98.5**	93.2	93.7	92.5	1.8	0.2	0.1
LV-Facebook	**100.0**	99.7	99.8	99.8	**100.0**	**100.0**	**100.0**
LV-YouTube	**100.0**	**100.0**	99.9	**100.0**	**100.0**	99.0	98.4
SBV-Instagram	**89.7**	74.7	76.5	78.8	10.0	6.3	6.4
SBV-TikTok	**93.3**	81.8	81.8	76.3	38.3	34.3	34.5
SBV-VkClips	**95.7**	94.0	91.3	92.4	53.2	37.7	46.0
SBV-YouTube	**98.2**	96.6	94.7	96.4	1.1	0.2	0.2
BV-Facebook	**87.7**	78.2	79.7	77.6	5.6	3.2	3.8
BV-Kinopoisk	**94.1**	84.1	85.8	89.8	5.4	4.0	4.1
BV-Netflix	**98.5**	97.2	95.2	93.7	50.7	52.3	56.1
BV-PrimeVideo	**91.3**	86.7	84.1	84.7	32.5	24.7	26.8
BV-Vimeo	**94.8**	90.5	90.2	81.4	72.0	19.5	68.6
BV-VkVideo	**88.6**	80.5	80.4	79.7	10.5	0.0	0.1
BV-YouTube	**85.9**	84.3	77.0	78.5	22.3	19.6	20.2
Web (known)	**99.7**	99.5	99.4	99.4	98.0	98.0	98.0
Macro-F-score (average)	**94.6**	89.9	88.7	88.9	38.4	31.4	35.1

LV is Live Video, (S)BV is (Short) Buffered Video, and BA is Buffered Audio.

paper, and the baselines, hC4.5 and UW, we compare them to the packet-based algorithms on the entire dataset. To verify that packet-based features still retain some informational value compared to flow-based features, we include a flow-based classifier, CESNET, as a baseline. Table 11 presents the obtained results.

The results show that hRFTC outperforms the best existing packet-based TC algorithms (RB-RF, MATEC, and BGRUA) by 56.3%, 63.3%, and 59.6%, respectively, in terms of macro F-score. Furthermore, hRFTC reduces the inaccuracy of the best-known hybrid classifiers, approximately, by half. This indicates the high efficiency of the proposed set of flow-based and packet-based features as compared with the existing algorithms. For example, the number and sizes of Encrypted Extensions and Certificate Chains in TLS messages can vary depending on the server and can partially reveal information about the traffic types. As a result, various servers produce different sizes of TLS messages, which reveal the servers and traffic types and remain unaffected by ECH. Additionally, the size of the client response that concludes the handshake and the sizes of the client requests are also unaffected by ECH.

Although the UW algorithm slightly outperforms hC4.5, it achieves much lower TC quality compared with the results reported in the original paper. We attribute this performance degradation to the accurate ECH simulation considered in our paper. As discussed in Section II, most papers on encrypted TC only hide the SNI value in the CH from the TC algorithm but do not hide its length. However, this feature can easily reveal the SNI, which is addressed in the ECH specification with padding recommendations [21]. Furthermore, to ensure eTC, we consider fewer packets for flow-based features, which also leads to performance degradation. The same conclusion holds for the flow-based CESNET algorithm. Additionally, the results of CESNET, which are lower than hRFTC and UW, confirm that TLS metadata is still useful for TC, and hybrid eTC algorithms can achieve higher TC quality than flow-based ones.

D. IMPACT OF TLS AND STATISTICAL FEATURES

A key feature of hRFTC is its interpretability. We evaluate the impact of the payload-based and flow-based features with the help of the Gini-Impurity-based (GI) feature importance metric [82].

Table 12 presents the top 30 features normalized by the maximal observed value. It illustrates that both types of features play an important role in the classification process, but the flow-based features have more impact. The reason for the low impact of packet-based features is the limited distinction in TLS setups: the average distinction metric value across the entire dataset is only 0.1753. Additionally, the diversity of the dataset in collection time and locations boosts the relevance of size-related flow-based features over time-related ones.

However, the top 30 most important features still contain some UL and DL IPT statistics. Hence, we can conclude

TABLE 12. hRFTC: the Gini-impurity-based feature importance normalized by the maximal observed value.

Rank	Feature	Impurity-based Feature Importance
1	CH Cipher Suites length	1.00
2	DL PSs Cumulative Sum	0.62
3	DL PSs Sorted Unique #1	0.50
4	UL PSs Std	0.46
5	UL PSs Cumulative Sum	0.45
6	UL PSs #2	0.44
7	DL IPTs Sum	0.43
8	DL PSs 25th percentile	0.43
9	DL PSs average	0.42
10	UL PSs 75th percentile	0.41
11	SH Cipher Suite	0.40
12	DL PSs Std	0.40
13	UL PSs Sorted Unique #2	0.40
14	UL PSs max	0.38
15	DL PSs 50th percentile	0.37
16	UL PSs average	0.36
17	DL PSs 75th percentile	0.33
18	UL IPTs Sum	0.28
19	UL PS #1 (CH length)	0.27
20	DL PS #2	0.26
21	UL IPTs min	0.26
22	UL PS 750-1000B freq	0.25
23	UL PSs Sorted Unique #3	0.25
24	CH Extensions Length	0.24
25	SH Extension Type #2	0.23
26	UL IPTs 25th percentile	0.23
27	UL IPTs 50th percentile	0.22
28	SH Extension Type #1	0.22
29	DL PS #3	0.21
30	UL IPTs max	0.2

TABLE 13. hRFTC: Sum of GI values over payload, PS, and IPT features normalized by the sum of all GI values.

Payload		IPT	PS			
CH	SH	IPT Stats	PS Hist	PSs Unique	PS Stats	PS Pattern
0.18	0.09	0.16	0.05	0.11	0.16	0.26
0.27		0.16	0.57			

that all the considered classes of features, namely, TLS parameters, PS statistics, and IPT statistics contribute noticeably to the classification quality. To check this observation, we evaluate the total impact of different groups of features on TC quality. Namely, we consider CH and SH (payload) features, IPT stats, PS histogram, PSs unique, PS stats, and PS pattern. We normalize the sum of their GI values by the sum of all GI values to reflect the relative impact.

Table 13 shows the obtained distribution of importance. We observe the following. First, the most valuable features are related to the packet sizes. They contribute more than 50% of total importance. Second, despite ECH and other efforts to encrypt as much of the TLS messages as possible, payload features still accumulate around 30% of importance.

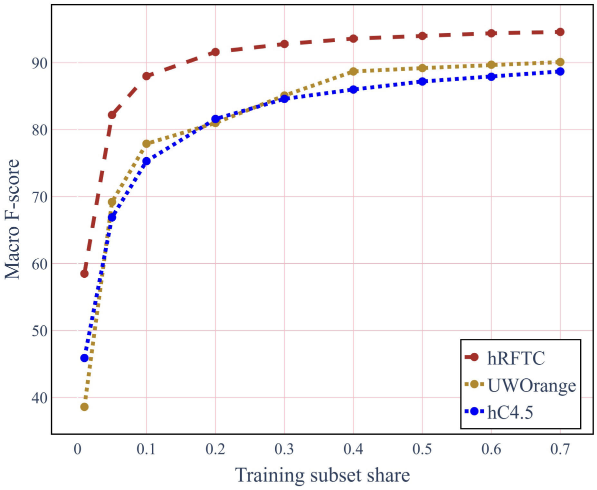

FIGURE 4. F-score depending on the training subset share.

TABLE 14. TC quality depending on training locations.

Test Country	Share in Dataset	Training Country	Classifier Macro F-score [%]		
			hRFTC	hC4.5	UW
Germany	18.8%	Others	38.4	26.9	19.5
Kazakhstan	3.0%	Others	57.3	32.3	27.5
Russia	29.2%	Others	49.8	35.6	20.9
Spain	16.3%	Others	38.5	34.4	12.6
Turkey	25.2%	Others	35.1	26.0	16.4
USA	7.5%	Others	49.2	41.4	21.3

According to the results presented in Table 11 the lowest F-score is achieved for QUIC traffic generated by BV-YouTube, BV-Facebook, and SBV-Instagram. Meanwhile, the feature-importance study reveals the reason for that. According to Table 13, the packet size features contribute a lot to the accuracy of the classification. However, with QUIC PADDING, all QUIC handshake packets have the same lengths. These results show that QUIC PADDING frames, indeed, complicate the TC, although, their impact is rather limited.

E. GENERALIZATION ABILITY OF HYBRID CLASSIFIERS

In a real-world implementation, the volume of traffic analyzed by a classifier surpasses the amount of training data rather fast and the application patterns can change over time. Therefore, in this Section, we study the generalization capabilities of the hybrid algorithms by investigating the impact of training dataset size and the dataset collection location on the classification quality of the eTC algorithms.

In the first experiment, we gradually reduce the training subset share from the default 70% to 1% of the collected dataset and compare the macro F-scores of hRFTC, UW, and hC4.5. Fig. 4 demonstrates the exceptional generalization ability of the developed hRFTC algorithm. The reduction of the training subset from 70% to 10% of the dataset leads to a mere 7% reduction in the macro F-score. Moreover, when training on 10% of the dataset, the hRFTC algorithm achieves approximately the same macro F-score as the hC4.5 algorithm trained at 70% of the dataset and the UW trained at 40%.

In the second experiment, we study how well the algorithms can classify traffic from previously unseen geographic locations. For that, we put all traffic from a specific country into the test set and traffic from all other countries into the training set. Table 14 compares the algorithms' performance across different test countries. For each country, we indicate the proportion of flows originating from that country within the entire dataset.

The results show that, despite the exceptional generalization capabilities of hRFTC, the classifier performs poorly in a new geographic location, similar to other hybrid baselines. Notably, as Fig. 4 and Table 14 indicate, for each of the locations, the size of the training dataset is more than enough for the algorithms to learn well during training. However, it did not help them to achieve good TC quality for the traffic of an unseen location. This highlights that the classifiers are effective within known geographic regions but struggle with unseen locations. Furthermore, although packet IPTs are strongly location-dependent, the results show that packet-size-related features, which significantly impact TC quality (see Table 13), also vary by location.

We see the following possible reasons for such an effect. First, depending on the last-mile network type and the network path between the client and the server, the TCP maximum segment size may vary. Therefore, the same application-layer data can be fragmented differently, affecting the PS and IPT statistics analyzed by the TC algorithm. Second, many companies use various content delivery networks (CDNs) to minimize the response and data delivery times for their clients. The settings of a server and, thereby, the lengths of the first packets of the flow can depend on a CDN provider responsible for the delivery of the data. Moreover, the choice of the CDN provider for a particular client depends on several factors, including the client's geolocation and the current load of different CDNs [83].

Therefore, from the obtained results, we can conclude that to achieve high classification quality, modern eTC algorithms based on flow-based features should be trained in the same geographical location where they will be used.

VII. CONCLUSION

In this paper, we have addressed the problem of early Traffic Classification of TLS flows with Encrypted ClientHello. We collected a vast dataset of TLS-encrypted traffic in various countries in North America, Europe, and Asia, split the flows into 19 classes by service requirements, and simulated the ClientHello encryption on it. Using the TLS distinction metric, we demonstrated that many service providers utilize similar TLS preferences from the server side. Thus, even the best state-of-the-art classification algorithms that rely only on TLS metadata perform poorly in the ECH scenario. Then, we designed the hRFTC algorithm, that outperforms the best state-of-the-art eTC algorithms. We demonstrated the excellent generalization ability of the proposed hRFTC: it performs well even when training only on 10% of the dataset. Finally, we showed that classifying traffic observed at a particular geographic location requires the eTC algorithm to be trained on the traffic from that location.

We see a few open problems following the results presented in the paper. First, remains unexplored the magnitude of QoS degradation due to the remaining inaccuracy of the state-of-the-art eTC algorithms. Second, QUIC protocol allows for connection migration in case of, e.g., an endpoint changing its IP address [70]. Such an event would require the middleboxes on the new route to classify the flow by its mid-flight packets, which, e.g., do not include the TLS handshake. In that case, the attainable accuracy of TC is of question. Third, peer-to-peer traffic significantly contributes to the overall Internet traffic. Its classification has not yet been considered in the ECH scenario [84], [85], and therefore, the applicability and accuracy of the existing eTC algorithms are unclear. Finally, many modern web applications initiate multiple flows within a short time. For example, web pages download fonts, scripts, stylesheets, etc.; video streaming downloads advertisements and posters. Therefore, the network flow appearing within proximity can reveal some information about each other. How to employ such correlations to further improve eTC remains another intriguing question.

REFERENCES

[1] C. Lin, K. Wang, and G. Deng, "A QoS-aware routing in SDN hybrid networks," *Proc. Comput. Sci.*, vol. 110, pp. 242–249, Jan. 2017.

[2] S. Rajasekhar, I. Khalil, and Z. Tari, "Probabilistic QoS routing inWiFi P2P networks," in *Proc. 20th Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, vol. 1, Apr. 2006, pp. 811–816.

[3] H. Huang, S. Guo, P. Li, B. Ye, and I. Stojmenovic, "Joint optimization of rule placement and traffic engineering for QoS provisioning in software defined network," *IEEE Trans. Comput.*, vol. 64, no. 12, pp. 3488–3499, Dec. 2015.

[4] M. Liubogoshchev, D. Zudin, A. Krasilov, A. Krotov, and E. Khorov, "DeSlice: An architecture for QoE-aware and isolated RAN slicing," *Sensors*, vol. 23, no. 9, p. 4351, Apr. 2023.

[5] S. Budhkar and V. Tamarapalli, "An overlay management strategy to improve QoS in CDN-P2P live streaming systems," *Peer Peer Netw. Appl.*, vol. 13, no. 1, pp. 190–206, Jan. 2020.

[6] Q. Wu, Q. Liu, Z. Jia, N. Xin, and T. Chen, "P4SQA: A P4 switch-based QoS assurance mechanism for SDN," *IEEE Trans. Netw. Service Manage.*, vol. 20, no. 4, pp. 4875–4886, Dec. 2023.

[7] P. Gou and C. Zheng, "Improved multi-granularity cascade forest network traffic classification optimization for SDN-based differentiated QoS guarantees," in *Proc. 3rd Int. Conf. Electron. Inf. Eng. Comput. Commun. (EIECC)*, vol. 161, Dec. 2023, pp. 455–459.

[8] I. F. Akyildiz, E. Khorov, A. Kiryanov, D. Kovkov, A. Krasilov, M. Liubogoshchev, D. Shmelkin, and S. Tang, "XStream: A new platform enabling communication between applications and the 5G network," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2018, pp. 1–6.

[9] *Information Technology—Dynamic Adaptive Streaming Over HTTP(DASH)—Part 5: Server and Network Assisted DASH (SAND)*, document ISO/IEC 23009-5:2017, 2017.

[10] *Northbound Application Programming Interface (API) for Multimedia Broadcast/Multicast Service (MBMS) at the xMB Reference Point; (Release 16)*, Standard 3GPP, TS 26.348, 2020.

[11] M. Abbasi, A. Shahraki, and A. Taherkordi, "Deep learning for network traffic monitoring and analysis (NTMA): A survey," *Comput. Commun.*, vol. 170, pp. 19–41, Mar. 2021.

[12] *HTTParchive*. Accessed: Feb. 15, 2023. [Online]. Available: https://httparchive.org/reports/state-of-the-web#pctHttps

[13] S. Rezaei, B. Kroencke, and X. Liu, "Large-scale mobile app identification using deep learning," *IEEE Access*, vol. 8, pp. 348–362, 2020.

[14] A. Dainotti, A. Pescape, and C. Sansone, "Early classification of network traffic through multi-classification," in *Proc. Int. Workshop Traffic Monit. Anal.* Berlin, Germany: Springer, 2011, pp. 122–135.

[15] S. Paramasivam and R. L. Velusamy, "Cor-ENTC: Correlation with ensembled approach for network traffic classification using SDN technology for future networks," *J. Supercomput.*, vol. 79, no. 8, pp. 8513–8537, May 2023.

[16] R. Jayaraman, B. Manickam, S. Annamalai, M. Kumar, A. Mishra, and R. Shrestha, "Effective resource allocation technique to improve QoS in 5G wireless network," *Electronics*, vol. 12, no. 2, p. 451, Jan. 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/2/451

[17] A. Kurapov, D. Shamsimukhametov, M. Liubogoshchev, and E. Khorov, "CloudETC: A privacy-preserving encrypted traffic classification platform for QoS in Wi-Fi," in *Proc. IEEE Int. Black Sea Conf. Commun. Netw. (BlackSeaCom)*, Jul. 2023, pp. 244–246.

[18] S. Rezaei and X. Liu, "Deep learning for encrypted traffic classification: An overview," *IEEE Commun. Mag.*, vol. 57, no. 5, pp. 76–81, May 2019.

[19] E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.3*, document RFC 8446, Aug. 2018. Accessed: Feb. 12023. [Online]. Available: https://tools.ietf.org/html/rfc8446

[20] D. Shamsimukhametov, M. Liubogoshchev, E. Khorov, and I. F. Akyldiz, "Are neural networks the best way for encrypted traffic classification?" in *Proc. Int. Conf. Eng. Telecommun. (En&T)*, Nov. 2021, pp. 1–5.

[21] E. Rescorla, K. Oku, N. Sullivan, and C. A. Wood. (Mar. 2024). *TLS Encrypted Client Hello*. Internet Engineering Task Force, Internet-Draft Draftietf-TLS-ESNI-18, Work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-tls-esni/18/

[22] D. Benjamin, *Applying Generate Random Extensions and Sustain Extensibility (Grease) to TLS Extensibility*, document RFC 8701, RFC Editor, Internet Requests for Comments, Wilmington, DE, USA, Jan. 2020. Accessed: Jul. 15, 2022. [Online]. Available: https://www.rfc-editor.org/info/rfc8701

[23] IANA. (2022). *Transport Layer Security (TLS) Extensions*. Accessed: Feb. 1, 2024. [Online]. Available: https://www.iana.org/assignments/tls-extensiontype-values/tls-extensiontype-values.xml

[24] D. Shamsimukhametov, A. Kurapov, M. Liubogoshchev, and E. Khorov, "Is encrypted ClientHello a challenge for traffic classification?" *IEEE Access*, vol. 10, pp. 77883–77897, 2022.

[25] N. Malekghaini, E. Akbari, M. A. Salahuddin, N. Limam, R. Boutaba, B. Mathieu, S. Moteau, and S. Tuffin, "AutoML4ETC: Automated neural architecture search for real-world encrypted traffic classification," *IEEE Trans. Netw. Service Manage.*, vol. 21, no. 3, pp. 2715–2730, Jun. 2024.

[26] *ISCX VPN-nonVPN Encrypted Network Traffic Dataset*. Accessed: Feb. 15, 2023. [Online]. Available: https://www.unb.ca/cic/datasets/vpn.html

[27] R. Wang, Z. Liu, Y. Cai, D. Tang, J. Yang, and Z. Yang, "Benchmark data for mobile app traffic research," in *Proc. 15th EAI Int. Conf. Mobile Ubiquitous Syst., Comput., Netw. Services*, Nov. 2018, pp. 402–411.

[28] J. F. Shbair, T. Cholez, J. Francois, and I. Chrisment. (2016). *HTTPS Websites Dataset*. [Online]. Available: http://betternet.lhs.loria.fr/datasets/https/

[29] D. Shamsimukhametov, M. Liubogoshchev, E. Khorov, and I. Akyildiz, "YouTube Netflix web dataset for encrypted traffic classification," in *Proc. Int. Conf. Eng. Telecommun.*, 2021, pp. 1–5.

[30] J. Luxemburk, K. Hynek, T. Čejka, A. Lukačovič, and P. Šiška, "CESNET-QUIC22: A large one-month QUIC network traffic dataset from backbone lines," *Data Brief*, vol. 46, Feb. 2023, Art. no. 108888.

[31] I. Akbari, M. A. Salahuddin, L. Ven, N. Limam, R. Boutaba, B. Mathieu, S. Moteau, and S. Tuffin, "A look behind the curtain: Traffic classification in an increasingly encrypted Web," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 49, no. 1, pp. 23–24, Jun. 2021.

[32] X. Liu, J. You, Y. Wu, T. Li, L. Li, Z. Zhang, and J. Ge, "Attention-based bidirectional GRU networks for efficient HTTPS traffic classification," *Inf. Sci.*, vol. 541, pp. 297–315, Dec. 2020.

[33] J. Cheng, Y. Wu, Y. E, J. You, T. Li, H. Li, and J. Ge, "MATEC: A lightweight neural network for online encrypted traffic classification," *Comput. Netw.*, vol. 199, Nov. 2021, Art. no. 108472.

[34] W. M. Shbair, T. Cholez, J. Francois, and I. Chrisment, "Early identification of services in HTTPS traffic," 2020, *arXiv:2008.08350*.

[35] N. Malekghaini, E. Akbari, M. A. Salahuddin, N. Limam, R. Boutaba, B. Mathieu, S. Moteau, and S. Tuffin, "Deep learning for encrypted traffic classification in the face of data drift: An empirical study," *Comput. Netw.*, vol. 225, Apr. 2023, Art. no. 109648.

[36] Z. Wang, "The applications of deep learning on traffic identification," *BlackHat USA*, vol. 24, no. 11, pp. 1–10, 2015.

[37] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2017, pp. 712–717.

[38] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," in *Proc. IEEE Int. Conf. Intell. Secur. Informat. (ISI)*, Jul. 2017, pp. 43–48.

[39] Z. Zou, J. Ge, H. Zheng, Y. Wu, C. Han, and Z. Yao, "Encrypted traffic classification with a convolutional long short-term memory neural network," in *Proc. IEEE 20th Int. Conf. High Perform. Comput. Commun., IEEE 16th Int. Conf. Smart City; IEEE 4th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Jun. 2018, pp. 329–334.

[40] S. Cui, B. Jiang, Z. Cai, Z. Lu, S. Liu, and J. Liu, "A session-packets-based encrypted traffic classification using capsule neural networks," in *Proc. IEEE 21st Int. Conf. High Perform. Comput. Commun.; IEEE 17th Int. Conf. Smart City; IEEE 5th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Aug. 2019, pp. 429–436.

[41] H. Yao, C. Liu, P. Zhang, S. Wu, C. Jiang, and S. Yu, "Identification of encrypted traffic through attention mechanism based long short term memory," *IEEE Trans. Big Data*, vol. 8, no. 1, pp. 241–252, Feb. 2022.

[42] W. Zheng, J. Zhong, Q. Zhang, and G. Zhao, "MTT: An efficient model for encrypted network traffic classification using multi-task transformer," *Appl. Intell.*, vol. 52, no. 9, pp. 10741–10756, Jul. 2022.

[43] Y. Xu, J. Cao, K. Song, Q. Xiang, and G. Cheng, "FastTraffic: A lightweight method for encrypted traffic fast classification," *Comput. Netw.*, vol. 235, Nov. 2023, Art. no. 109965.

[44] M. Shen, K. Ye, X. Liu, L. Zhu, J. Kang, S. Yu, Q. Li, and K. Xu, "Machine learning-powered encrypted network traffic analysis: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 791–824, 1st Quart., 2023.

[45] W. M. Shbair, T. Cholez, J. Francois, and I. Chrisment, "A multi-level framework to identify HTTPS services," in *Proc. NOMS IEEE/IFIP Netw. Operations Manage. Symp.*, Apr. 2016, pp. 240–248.

[46] C. Ma, X. Du, and L. Cao, "Improved KNN algorithm for fine-grained classification of encrypted network flow," *Electronics*, vol. 9, no. 2, p. 324, Feb. 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/2/324

[47] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of encrypted and VPN traffic using time-related features," in *Proc. 2nd Int. Conf. Inf. Syst. Secur. Privacy*, 2016, pp. 407–414.

[48] M. Jaber, R. G. Cascella, and C. Barakat, "Can we trust the inter-packet time for traffic classification?" in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2011, pp. 1–5.

[49] Y. Kumano, S. Ata, N. Nakamura, Y. Nakahira, and I. Oka, "Towards real-time processing for application identification of encrypted traffic," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2014, pp. 136–140.

[50] C. Wang, T. Xu, and X. Qin, "Network traffic classification with improved random forest," in *Proc. 11th Int. Conf. Comput. Intell. Secur. (CIS)*, Dec. 2015, pp. 78–81.

[51] M. Chari, H. Srinidhi, and T. E. Somu, "Network traffic classification by packet length signature extraction," in *Proc. IEEE Int. WIE Conf. Electr. Comput. Eng. (WIECON-ECE)*, Nov. 2019, pp. 1–4.

[52] M. Dener, S. Al, and G. Ok, "RFSE-GRU: Data balanced classification model for mobile encrypted traffic in big data environment," *IEEE Access*, vol. 11, pp. 21831–21847, 2023.

[53] F. Zaki, F. Afifi, S. A. Razak, A. Gani, and N. B. Anuar, "GRAIN: Granular multi-label encrypted traffic classification using classifier chain," *Comput. Netw.*, vol. 213, Aug. 2022, Art. no. 109084.

[54] L. Vu, C. T. Bui, and Q. U. Nguyen, "A deep learning based method for handling imbalanced problem in network traffic classification," in *Proc. 8th Int. Symp. Inf. Commun. Technol.*, Dec. 2017, pp. 333–339.

[55] Z. Chen, K. He, J. Li, and Y. Geng, "Seq2Img: A sequence-to-image based approach towards IP traffic classification using convolutional neural networks," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 1271–1276.

[56] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Network traffic classifier with convolutional and recurrent neural networks for Internet of Things," *IEEE Access*, vol. 5, pp. 18042–18050, 2017.

[57] M. H. Pathmaperuma, Y. Rahulamathavan, S. Dogan, and A. M. Kondoz, ‘‘Deep learning for encrypted traffic classification and unknown data detection,’’ *Sensors*, vol. 22, no. 19, p. 7643, Oct. 2022.

[58] Y. Wang, H. He, Y. Lai, and A. X. Liu, ‘‘A two-phase approach to fast and accurate classification of encrypted traffic,’’ *IEEE/ACM Trans. Netw.*, vol. 31, no. 3, pp. 1–16, Sep. 2022.

[59] M. Shen, J. Zhang, L. Zhu, K. Xu, and X. Du, ‘‘Accurate decentralized application identification via encrypted traffic analysis using graph neural networks,’’ *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2367–2380, 2021.

[60] S.-J. Xu, G.-G. Geng, X.-B. Jin, D.-J. Liu, and J. Weng, ‘‘Seeing traffic paths: Encrypted traffic classification with path signature features,’’ *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 2166–2181, 2022.

[61] S. Roy, T. Shapira, and Y. Shavitt, ‘‘Fast and lean encrypted internet traffic classification,’’ *Comput. Commun.*, vol. 186, pp. 166–173, Mar. 2022.

[62] A. Rasteh, F. Delpech, C. Aguilar-Melchor, R. Zimmer, S. B. Shouraki, and T. Masquelier, ‘‘Encrypted Internet traffic classification using a supervised spiking neural network,’’ *Neurocomputing*, vol. 503, pp. 272–282, Sep. 2022.

[63] J. Luxemburk and T. Čejka, ‘‘Fine-grained TLS services classification with reject option,’’ *Comput. Netw.*, vol. 220, Jan. 2023, Art. no. 109467.

[64] P. Lin, K. Ye, Y. Hu, Y. Lin, and C.-Z. Xu, ‘‘A novel multimodal deep learning framework for encrypted traffic classification,’’ *IEEE/ACM Trans. Netw.*, vol. 31, no. 3, pp. 1–16, Oct. 2022.

[65] B. Anderson and D. McGrew, ‘‘Machine learning for encrypted malware traffic classification: Accounting for noisy labels and non-stationarity,’’ in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 1723–1732.

[66] X. Lin, G. Xiong, G. Gou, Z. Li, J. Shi, and J. Yu, ‘‘ET-BERT: A contextualized datagram representation with pre-training transformers for encrypted traffic classification,’’ in *Proc. ACM Web Conf.*, Apr. 2022, pp. 633–642.

[67] R. Barik, M. Welzl, A. Elmokashfi, T. Dreibholz, S. Islam, and S. Gjessing, ‘‘On the utility of unregulated IP DiffServ code point (DSCP) usage by end systems,’’ *Perform. Eval.*, vol. 135, Nov. 2019, Art. no. 102036.

[68] *Examining Http/3 Usage One Year on*. Accessed: May 27, 2024. [Online]. Available: https://blog.cloudflare.com/http3-usage-one-year-on/

[69] M. Thomson and S. Turner, *Using TLS to Secure QUIC*, document RFC 9001, IETF, Request for Comments, 9001.

[70] J. Iyengar and M. Thomson, *QUIC: A UDP-Based Multiplexed and Secure Transport*, document RFC 9000, May 2021. [Online]. Available: https://www.rfc-editor.org/info/rfc9000

[71] S. Siby, L. Barman, C. Wood, M. Fayed, N. Sullivan, and C. Troncoso, ‘‘You get PADDING, everybody gets PADDING! You get privacy? Evaluating practical QUIC website fingerprinting protections for the masses,’’ 2022, *arXiv:2203.07806*.

[72] K. Park and W. Willinger, ‘‘Self-similar network traffic: An overview,’’ in *Self-Similar Network Traffic and Performance Evaluation*. Hoboken, NJ, USA: Wiley, 2000, pp. 1–38.

[73] *Wireshark*. Accessed: Oct. 1, 2024. [Online]. Available: https://www.wireshark.org/

[74] (2023). *The Global Internet Phenomena Report Jauary 2023*. [Online]. Available: https://www.sandvine.com/global-internet-phenomena-report-2023

[75] *Alexa 1M, Top Visited Webcites*. Accessed: Feb. 15, 2023. [Online]. Available: http://s3.amazonaws.com/alexa-static/top-1m.csv.zip

[76] *WebDriver, S.* Accessed: Oct. 1, 2024. [Online]. Available: https://www.selenium.dev/

[77] *Websites Using Quic*. Accessed: Nov. 19, 2022. [Online]. Available: https://trends.builtwith.com/websitelist/QUIC

[78] Z. Tsiatsikas, G. Karopoulos, and G. Kambourakis, ‘‘Measuring the adoption of TLS encrypted client hello extension and its forebear in the wild,’’ in *Proc. ESORICS*. Cham, Switzerland: Springer, 2023, pp. 177–190.

[79] J. Zirngibl, F. Gebauer, P. Sattler, M. Sosnowski, and G. Carle, ‘‘QUIC hunter: Finding QUIC deployments and identifying server libraries across the Internet,’’ 2023, *arXiv:2308.15841*.

[80] D. Powers, ‘‘Evaluation: From precision, recall and F-factor to ROC, informedness, markedness & correlation,’’ *Mach. Learn. Technol.*, vol. 2, pp. 37–63, Jan. 2008.

[81] D. R. Shamsimukhametov, A. A. Kurapov, M. V. Liubogoshchev, and E. M. Khorov, ‘‘Indistinguishability of traffic by open TLS parameters with encrypted ClientHello,’’ *J. Commun. Technol. Electron.*, vol. 68, no. 12, pp. 1523–1529, Dec. 2023.

[82] L. Rokach and O. Maimon, ‘‘Top-down induction of decision trees classifiers—A survey,’’ *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, vol. 35, no. 4, pp. 476–487, Nov. 2005.

[83] S. Cui, M. R. Asghar, and G. Russello, ‘‘Multi-CDN: Towards privacy in content delivery networks,’’ *IEEE Trans. Depend. Secure Comput.*, vol. 17, no. 5, pp. 984–999, Sep. 2020.

[84] M. Bhatia, V. Sharma, P. Singh, and M. Masud, ‘‘Multi-level P2P traffic classification using heuristic and statistical-based techniques: A hybrid approach,’’ *Symmetry*, vol. 12, no. 12, p. 2117, Dec. 2020. [Online]. Available: https://www.mdpi.com/2073-8994/12/12/2117

[85] M. S. A. Ansari, K. Pal, and M. C. Govil, ‘‘Revisiting of peer-to-peer traffic: Taxonomy, applications, identification techniques, new trends and challenges,’’ *Knowl. Inf. Syst.*, vol. 65, no. 11, pp. 4479–4536, Nov. 2023.

DANIL SHAMSIMUKHAMETOV received the B.S. and M.S. degrees in applied mathematics and physics from Moscow Institute of Physics and Technology, Moscow, Russia, in 2020 and 2022, respectively, where he is currently pursuing the Ph.D. degree in telecommunications. He has been a Researcher with the Wireless Networks Laboratory, Institute for Information Transmission Problems of the Russian Academy of Sciences, since 2018. He participates in national projects and does research within the framework of joint research projects with the leading telecommunication companies. His research interests include machine learning applications in wireless systems and traffic classification.

ANTON KURAPOV (Member, IEEE) received the B.S. and M.S. degrees in applied mathematics and physics from Moscow Institute of Physics and Technology, Moscow, Russia, in 2022 and 2024, respectively. He has been a Researcher with the Wireless Networks Laboratory, Institute for Information Transmission Problems of the Russian Academy of Sciences, since 2020. His research interests include machine learning applications in wireless systems and traffic classification.

MIKHAIL LIUBOGOSHCHEV received the Ph.D. degree in telecommunications from Moscow Institute of Physics and Technology, in 2023. He has been a Researcher with the Network Protocols Research Laboratory and the Wireless Networks Laboratory, Institute for Information Transmission Problems of the Russian Academy of Sciences, since 2016 and 2018, respectively. His research interests include devoted to 5G and beyond wireless systems, QoS-aware cross-layer optimization, and stochastic network modeling and optimization.

EVGENY KHOROV (Senior Member, IEEE) received the D.Sc. degree. He is the Head of the Wireless Networks Laboratory, Institute for Information Transmission Problems of the Russian Academy of Sciences. He has led dozens of projects and authored over 200 articles on 5G, next-generation Wi-Fi, protocol design, and QoS-aware cross-layer optimization. Being a Voting Member of IEEE 802.11, he has contributed to IEEE 802.11ax. He was a recipient of multiple national and international scientific awards. In 2020, he was awarded as the Editor of the Year of *Ad Hoc Networks*. He gave tutorials and participated in panels at large IEEE events. He is the Editor-in-Chief of *Problems of Information Transmission*.