Computer Networks 247 (2024) 110452

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Comparative analysis of DNS over HTTPS detectors

Kamil Jerabek [a], Karel Hynek [b,*], Ondrej Rysavy [a]

[a] Faculty of Information Technology, Brno University of Technology, Brno, Czechia
[b] CESNET z.s.p.o., Prague, Czechia

ARTICLE INFO

Keywords:
DNS over HTTPS
DoH
Detection
Comparative analysis
Machine learning
And network security

ABSTRACT

DNS over HTTPS (DoH) is a protocol that encrypts DNS traffic to improve user privacy and security. However, its use also poses challenges for network operators and security analysts who need to detect and monitor network traffic for security purposes. Therefore, there are multiple DoH detection proposals that leverage machine learning to identify DoH connections; however, these proposals were often tested on different datasets, and their evaluation methodologies were not consistent enough to allow direct performance comparison. In this study, seven DoH detection proposals were recreated and evaluated with six different experiments to answer research questions that targeted specific deployment scenarios concerning ML-model transferability, usability, and longevity. For thorough testing, a large Collection of DoH datasets along with a novel 5-week dataset was used, which enabled the evaluation of models' longevity. This study provides insights into the current state of DoH detection techniques and evaluates the models in scenarios that have not been previously tested. Therefore, this paper goes beyond classical replication studies and shows previously unknown properties of seven published DoH detectors.

1. Introduction

DNS over HTTPS (DoH) is becoming a recognized privacy-preserving technology that works on a simple concept of encapsulating DNS messages into an encrypted HTTPS channel [1]. The technology gained fast adoption among the service providers and also users [2]. It is nowadays used by default in popular web browsers [3], and it is already supported by major operating systems such as Windows [4] and MacOS [5]. Nevertheless, the fast DoH deployment still raises concerns mainly from the security community [6]. Plain-text DNS requests are essential in maintaining computer security since many intrusion detection and intrusion prevention systems rely on their payload inspection. Moreover, multiple parental control and policy enforcement in restricted company networks are based solely on domain name inspection. The DNS encryption by DoH thus bypasses the effective and timely-proven detection approach, leaving systems vulnerable to cybersecurity attacks that use DoH. Multiple examples of malware, exfiltration tools, and red-team attack vectors already exploit DoH for stealthy communication to avoid detection [6].

The popularity of DoH among threat actors can be partially attributed to its stealthiness. Compared to other encrypted DNS approaches, such as DNS over TLS or DNS over QUIC, DoH does not use dedicated ports; instead, it is designed to blend into other HTTPS traffic, leaving cybersecurity operators often unaware of its presence. Just the information about the presence of DoH in the network is valuable for network security operators to evaluate the risks. Detecting DoH is not a straightforward network security task. According to Garcia et al. [2], DoH detection based on blocklists of IP addresses and domain names is highly unreliable due to many small and private resolvers. Therefore, the researchers concentrated more on using encrypted traffic analysis techniques combined with machine learning (ML) to identify DoH connections in the traffic.

Over the past years, there have been multiple ML-based proposals of DoH presence detection, often surpassing the 99% detection accuracy. These novel ML-based DoH detectors have often been evaluated using different lab-created datasets (such as [7–9]) from a limited time period. However, without evaluation using the same data, the proposals cannot be compared; thus, the basic questions such as "*How effective are those approaches compared to each other?*" or "*How do the detectors behave in long-term?*" still lack their answers. Compared to other network security challenges [10–12], DoH detection task in network security is still missing a comparative study that would answer these questions. Therefore, this study is built on answering several defined research questions.

In this study, the previously published methodologies for DoH detection were followed and the detectors were recreated to evaluate their properties using the previously published Collection of datasets with DoH traffic [13]. Additionally, a new capture of real-world DoH traffic (using the same methodology as in the Collection of datasets) was

* Corresponding author.
E-mail addresses: ijerabek@fit.vutbr.cz (K. Jerabek), hynekkar@cesnet.cz (K. Hynek), rysavy@fit.vutbr.cz (O. Rysavy).

https://doi.org/10.1016/j.comnet.2024.110452
Received 23 January 2024; Received in revised form 5 April 2024; Accepted 20 April 2024
Available online 27 April 2024

created to study the pace of obsolescence of ML-based DoH detectors due to either data or concept drifts. These datasets allowed us to evaluate DoH detectors thoroughly and finally show the advantages and disadvantages of each DoH detection approach and provide valuable insights about the state of advancement in that field.

The contributions of this study may be summarized into the following points:

- Reimplementation of the existing flow-based DoH detection approaches and replication of the reported results on a common, more comprehensive dataset.
- Transferability evaluation of the DoH detectors between different networks.
- Creation of the Additional 5-week dataset to study the long-term usability of the DoH detectors.
- Long-term and short-term usability and performance degradation of DoH detectors.
- Resource consumption evaluation of the DoH detectors.

The rest of the manuscript is organized as follows: Section 2 summarizes related works. Section 3 summarizes the datasets used for the comparison. Section 4 provides the necessary background about DoH protocol and its detection possibilities. Section 5 briefly introduces the selected DoH detection approaches. Section 6 then defines the methodological approach. Section 7 describes the individual experiments and their results. Section 8 summarizes the observed properties across all the methods. Finally, Section 9 concludes this article.

2. Related works

Since DoH represents a significant change in domain name resolution, it has been thoroughly studied by multiple researchers from various perspectives. A survey performed by Hynek et al. [6] divided the research into four categories: (1) DoH performance measurements, (2) Research on DoH adoption, (3) Privacy-preserving research, and (4) DoH security research where the DoH detection research is identified as a subcategory. This comparative analysis thus falls into the fourth category.

The necessity of DoH recognition was first mentioned in 2020 by Bumglang et al. [14] in their survey concerning DoH mass deployment impact. The straightforward solution—a list of DoH resolvers' IP addresses was used by Bushart et al. [15] to recognize DoH connection in their traffic fingerprinting approach. However, in 2022, the comprehensiveness of publicly available lists of DoH resolvers was studied by Garcia et al. [2]. They performed an internet-wide scan to recognize DoH resolution capability. According to their results, IP-based DoH detection is inefficient due to the large number of private resolvers. The DoH resolvers operated by individuals are often not listed in public blocklists—the privately owned resolvers represent around 13% of all DoH resolvers on the Internet [2]. Moreover, they also showed significant changes in DoH resolvers IP addresses over time, leading to fast blocklist obsolescence.

Two studies published at a similar time by Vekhsin et al. [16] and MontazeriShatoori et al. [17] attempted to avoid IP-based DoH identification. Instead, they used distinctive DoH traffic shape [18] for its identification. Vekhsin et al. [16] used ipfixprobe[1] flow exporter, that extends flows for the first 30 individual packets—packet lengths, packet timestamps, TCP flags, and directions. From the first 30 individual packets, they extracted 18 discriminatory features that were used together with the AdaBoosted decision tree machine learning model. Their approach proved efficient in the DoH recognition task, achieving over 99% accuracy or 0.99 of F1 score.

Compared to Vekshin et al. [16], who extracted features from the first 30 packets, MontazeriShatoori et al. [17] created a novel flow exporter – DoHLyzer[2] – that was capable of extracting 28 traffic features calculated from the whole connection. They used the selected features together with multiple machine learning models, and the best one – Random Forest – achieved an F1 score of 0.993. Moreover, they also published the *CIRA-CIC-DoHBrw-2020* [7] dataset that they used in their study.

The following studies then continued to use the *CIRA-CIC-DoHBrw-2020* dataset in their DoH detection proposals. Banadaki et al. [19] achieved 100% using the *CIRA-CIC-DoHBrw-2020* dataset; however, his work was later severely criticized by Behnke et al. [20] because of the IP address presence in the feature vector resulting in reduced generalization of the model. By further exploring the feature vector, Behnke et al. [20] thus improved Banadaki's proposal. They removed the IP addresses, ports, and statistically insignificant features and finally trained and evaluated multiple machine learning models and achieved a high F1 of 0.998.

Following studies created by Casanova et al. [21], Jha et al. [22], Wu et al. [23], Zebin et al. [24] and Mitsuhashi et al. [25] also used data from *CIRA-CIC-DoHBrw-2020* dataset and achieved over 99% of accuracy. Nevertheless, most of these studies used features directly provided within the *CIRA-CIC-DoHBrw-2020*, which were seen as unpractical by Jerabek et al. [26]. They argue that some of the used features, such as median packet lengths, cannot be calculated streamwise and the whole packet series needs to be stored in the memory of the network monitoring device. This large memory requirement then highly limits the deployment into real-world monitoring infrastructure [26]. Therefore, Jerabek et al. [26] proposed a novel detector that uses only four features that can be easily extracted and calculated even from basic NetFlowV5. However, their approach should be combined with active probing verification mechanisms to ensure reliability. They evaluated their approach using the NetExP flow exportation tool[3] and custom dataset mixed with the *CIRA-CIC-DoHBrw-2020*. Their proposal achieved nearly *100*% of accuracy.

Even though there are many DoH detection proposals, and also DoH detection surveys [6,27] none of the previous studies attempted to compare the properties of each detector. Since the ML-based detection proposals used different datasets [7–9] their methodologies using a single common dataset were recreated and their experiments were replicated. This is the first DoH detection study that validates state-of-the-art approaches and assesses the state of the technology, its reliability (even in the face of drift), and deployment possibilities.

3. Used datasets

The comparative analysis uses the comprehensive collection of datasets with DoH traffic [13], which is provided in the form of anonymized packet captures (pcaps), thus allowing extraction of all features used by the detector proposals. The collection is divided into two dataset types: The Real-World and Generated. The Real-World traffic was captured on the CESNET2 network—a large Czech ISP. The real-world data contains ten days of DoH and HTTPS communication captured on backbone lines. The labels of the DoH connections are provided in the form of an IP list with the captured DoH resolvers.

The motivation behind the creation of additional generated data – the Generated datasets – was to enhance the comprehensiveness of the collection. The majority of the real-world DoH uses only a handful of resolvers (Google DNS and Cloudflare DNS [13]), which results in the underrepresentation of other resolvers in the dataset. The dataset was created in a virtual environment using docker, selenium, and two major web browsers (Firefox and Chrome) that were set to use DoH and send their requirements to 16 different resolvers. The exact methodology for

[1] https://github.com/CESNET/ipfixprobe.

[2] https://github.com/ahlashkari/DoHLyzer.git.

[3] https://github.com/kjerabek/netexp.

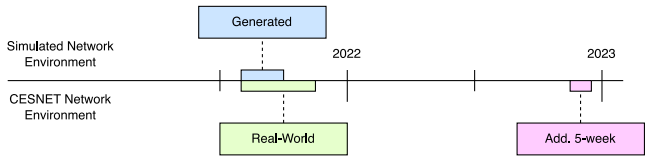

Fig. 1. Timeline of dataset capturing.

Table 1
Statistics about used types of dataset. Please note that the number of connections is not equal to the number of flows referred elsewhere in the work since each flow exporter maps connections to flows differently.

Dataset	Other connections	DoH connections	Size
Real-World	~156 000	~5 200 000	179 GB
Generated	~1 600 000	~346 000	250 GB
Add. 5-week	~17 000	~486 000	52 GB

generation and capturing Generated and Real-World data is described in detail in the data article [13].

Together, the collection of datasets provides a comprehensive set of DoH and regular HTTPS data. The Real-World part contains real-world timing and network characteristics, while the Generated part contains a large set of DoH resolvers, each with slightly different characteristics. According to the original data article [26], the two datasets are very different in terms of packet timing and size distribution. This paper is, however, not aiming to describe and study all the differences between Generated and Real-World datasets—all detailed differences are properly described in the original data article [13]. The differences between both datasets make the collection ideal for assessing the performance of models trained on lab-created data (Generated dataset) in the real-world environment.

Nevertheless, the collection of datasets does not allow us to experiment with drift—a phenomenon where the distribution of the input data changes over time, which causes the obsolescence of trained ML models. There have been numerous reports about ongoing drifts [28] and their impact on models' performance. To allow experiments concerning the longitudinal performance evaluation of the detectors, the additional dataset with captured data on the CESNET2 network has been created and published at the Zenodo platform [29]. The capturing of the novel dataset followed the same methodology as in the Real-World data part (as described in the data article [13]). The capture was automatically performed every Monday evening for five weeks between 28th of November till 26th of December 2022.

The information about all used datasets is provided in Table 1, and the times of dataset capturing are graphically shown in Fig. 1. Altogether, the used datasets contain more than seven million different flows making it the largest dataset used for DoH detector evaluation—it is seven times larger than the most extensive dataset *CIRA-CIC-DoHBrw-2020* used in related works. Moreover, almost half of the used dataset contains real-user traffic, ensuring a representative sample of real-world traffic traces. Lastly, the diversity of the traffic is ensured by the size of the CESNET2 ISP network, which is used by half a million users every day. Therefore, the dataset contains ~115 000 unique client public IP addresses and more than 10 000 unique server addresses.

4. Background on DoH detection

The DoH was standardized by IETF as RFC 8484 [1] in 2018 to enhance the privacy of domain name resolution and prevent on-path devices from interfering with DNS resolution process [1]. The RFC-compliant DoH protocol uses traditional DNS wireformat messages as defined in RFC 1035 [30] and sends them to the resolver as HTTP POST or GET requests. To identify DNS data inside HTTP, a client must provide HTTP `Content-Type` field with `application/dns-message` value. RFC 8484 also recommends using DoH only over HTTP/2 protocol due to its stream multiplexing feature that enables clients to send concurrent requests, making the resolution process much more efficient.

Despite the RFC recommendations, there are still resolvers on the internet that support only DoH over HTTP/1. Nevertheless, most browsers support DoH over HTTP/1 and overcome its request–response limitations by opening multiple concurrent TCP connections and multiplexing between them. Except for the novel headers, HTTP version, and possible cache-control options, RFC 8484 does not define any additional requirements and restrictions on top of the HTTP protocol. Such leeway in the specification allows resolvers and browsers to include various HTTP headers resulting in different packet sizes and thus in different traffic shapes, which can be used for client/resolver identification [16,31].

From the packet-level perspective, each DoH request or response is mapped into a single network packet [18], which defines the traffic shape of DoH connections. Apart from initial handshakes, both TLS and HTTP/2, the DoH connection is usually long-lasting and contains smaller packets that are sparsely distributed in the connections. Nevertheless, such traffic shape is also common for other HTTP-based APIs, and those are particularly challenging to distinguish from DoH.

The DoH detection proposals are often leveraging [6] the timing of packets, which is distinctive in the case of web-based DoH. A single web page visit often results in a burst of multiple DoH queries to domain names that host additional assets such as content delivery networks or javascript library providers [6,18,31]. The burstiness of the communication is the primary factor distinguishing DoH from other HTTP API calls [18]. Unfortunately, ML-based detection of short or single query DoH, which is often generated by malware, is still not solved [6] and can be recognized only by blocklists.

5. Selected DoH detection approaches

Several machine-learning approaches were published to distinguish DoH and regular HTTPS (non-DoH). Some of them focus only on the DoH detection task, while others include the task as the first stage, and in the second stage, they further focus on malicious DoH detection. Alternatively, some approaches focus solely on malicious DoH detection. Not all approaches could be selected in this comparative study, mainly due to their impossible reproducibility. This section provides a definition of selection criteria for DoH detection approaches to be considered in the comparative study and then briefly describes the selected approaches that are recreated within this study.

5.1. Definition of selection criteria

This study is limited to DoH detection proposals that utilize network flows as broadly accepted data sources in current monitoring and IDS solutions and further satisfy the following criteria to allow reproducibility and fair comparison:

PCAP processing: The proposal has to publish a used PCAP processing code or tool. Each PCAP processing pipeline is different and needs to be specified exactly to allow recreation of the approach.

Features: The proposal has to publish an exact definition of used detection features. Without the exact definition of the feature vector, the replication of the study is impossible.

No identifiers: The proposal must not use traffic identifiers such as IP addresses and ports as features. As discussed by Behnke et al. [20], these features would overfit the lab-created datasets, and make fair comparability with other approaches impossible.

Defined architecture: The proposal must precisely define the used classification algorithm and its parameters, such as layers in neural networks. Replication of proposal is impossible without the knowledge of the exact algorithm,

Table 2
The summary of criteria fulfillment across published DoH detection methods.

Paper	PCAP processing	Features	No identifiers	Defined architecture
MontazeriShatoori et al. [17]	✓	✓	✓	✓
Mitsuhashi et al. [25]	✓	✓	✓	✓
Behnke et al. [20]	✓	✓	✓	✓
Casanova et al. [21]	✓	✓	✓	✓
Zebin et al. [24]	✓	✓	✓	✓
Vekshin et al. [16]	✓	✓	✓	✓
Jerabek et al. [26]	✓	✓	✓	✓
Konopa et al. [32]	✗	✓	✓	✗
Jha et al. [22]	✓	✗	✓	✓
Banadaki [19]	✓	✓	✗	✓
Nguyen et al. [33]	✓	✓	✓	✗

Table 3
Selected DoH detection approach summary. The abbreviations stand for: **NF** - Number of required statistical features; **FE** - Flow elimination before passed to ML model.

Paper	NF	Tool	FE	Balancing	Scaling	Best algorithm	F1	Accuracy
MontazeriShatoori et al. [17]	28	DoHLyzer	NaN	–	–	Random forest	0.993	–
Mitsuhashi et al. [25]	28	DoHLyzer	NaN	–	–	XGBoost	0.998	99.8%
Behnke et al. [20]	26	DoHLyzer	NaN	–	–	Random forest	0.998	–
Casanova et al. [21]	28	DoHLyzer	NaN	Resampling	Min–max	BiLSTM	0.987[a]	99.0%
Zebin et al. [24]	29	DoHLyzer	NaN	SMOTE	Min–max	Balanced Stacked RF	0.999	99.98%
Vekshin et al. [16]	18	ipfixprobe	NaN + <5 pay pkts	SMOTE	–	Ada-oosted DT	0.976[a]	99.6%
Jerabek et al. [26]	4	NetExP	NaN + <120 pkts	–	Standard	XGBoost	0.998	99.9%

[a] Score is computed from the provided confusion matrix.

The list of all analyzed DoH detection approaches with the selected criteria fulfillment is written in Table 2. Studies of MontazeriShatoori et al. [17], Mitsuhashi et al. [25], Behnke et al. [20], Casanova et al. [21], Zebin et al. [24], Vekshin et al. [16] and Jerabek et al. [26] satisfied the conditions. Konopa et al. [32] did not include a tool or description of flow extraction from PCAP files, which makes the reproduction impossible. Moreover they also did not describe the used ML model architecture. Jha et al. [22] does not describe the exact set of features used for detection. Banadaki [19] used IP addresses or ports for DoH traffic classification. Lastly, the detector proposed by Nguyen et al. [33] could not be reproduced, due to the missing description of used classification architecture.

5.2. Brief overview of selected DoH detection approaches

Together, seven detection approaches were selected for recreation in the comparative study. This section provides a brief introduction to each one. Moreover, Table 3 summarizes the approaches, their performance metrics, and properties that were essential for their replication.

MontazeriShatoori et al. [17] was one of the first studies that published the DoH detection approach together with the dataset *CIRA-CIC-DoHBrw-2020* [7] that contained pre-extracted flow statistics by their DoHLyzer tool. They evaluated multiple ML algorithms but achieved the best F1 score of 0.993 with Random Forest and 28 features.

Mitsuhashi et al. [25] used the same 28 features extracted with the DoHLyzer tool as MontazeriShatoori. However, they utilized different algorithms, out of which the XGBoost with a maximum tree depth of 10 and the maximum number of bins set 1024 performed the best. They achieved F1 score of 0.998.

Behnke et al. [20] also used a feature vector exported by the DoHLyzer tool. Compared to Montazerishatoori et al. [17] and Mitsuhashi et al. [25], they removed two insignificant features and ended up with 26. Behnke et al. [20] achieved the best results with the Random Forest algorithm with a 0.998 F1 score. The only hyperparameter tuned was the maximum number of features considered in each split during training—they used 21 features as the optimal number.

Casanova et al. [21] also used the same 28 features as MontazeriShatoori. Nevertheless, they tuned Bi-directional LSTM for this task and achieved an accuracy of 99.0%. In their first study [21] from 2021, they described their algorithm design pipeline—they used min–max scaling and resampling of unbalanced data. Moreover, they showed the best hyperparameters for the achieved performance. However, the architecture of the neural network was missing until they published the following study [34] in 2023. The described architecture consists of 60 neurons of bi-directional LSTM that also achieved the best performance on *CIRA-CIC-DoHBrw-2020* dataset when compared to other different neural network architectures.

Zebin et al. [24] worked with 29 features extracted with the DoHLyzer tool. They used a min–max scaler and Synthetic Minority Oversampling Technique (SMOTE) to deal with imbalanced datasets. They proposed a stacking classifier approach with three random forests as base classifiers—each forest contained 10 trees and 28 features in each split. The results from the base classifiers are then used as features for Logistic Regression, which makes the final classification with an F1 score of 0.999.

The approaches mentioned above utilize the same dataset and features extracted by the DoHLyzer tool. Their primary focus was tuning different algorithms, normalization, and data balancing. Further approaches use different extraction tools, focusing more on data than on tuning the algorithms.

Vekshin et al. [16] used flows extended for the first 30 individual packets extracted with the ipfixprobe tool and further computing 18 discriminatory features. Apart from feature selection, their training pipeline also uses SMOTE, and they skip flows with less than five payload packets directly marking them as non-DoH. The algorithm that had the highest performance among the ones evaluated was an Ada-Boosted Decision Tree classifier with a maximal depth of a single tree set to 30 and the number of estimators set to 5, reaching 99.6% accuracy.

Jerabek et al. [26] used information from standard flow data (provided even by NetFlow V5) extracted by their experimental NetExP tool to compute four features. They show the results of all flow classifications, but they point out that the short flows should be omitted from the classification, and they further skip flows with less than 120 packets. They proposed a system for dealing with shorter flows that perform active verification. Out of all tested algorithms, they achieved the best results with the XGBoost algorithm and a 0.998 F1 score.

6. General methodology

The selected approaches differ in raw data processing, extracted features, algorithms used, and accompanied preprocessing. The common

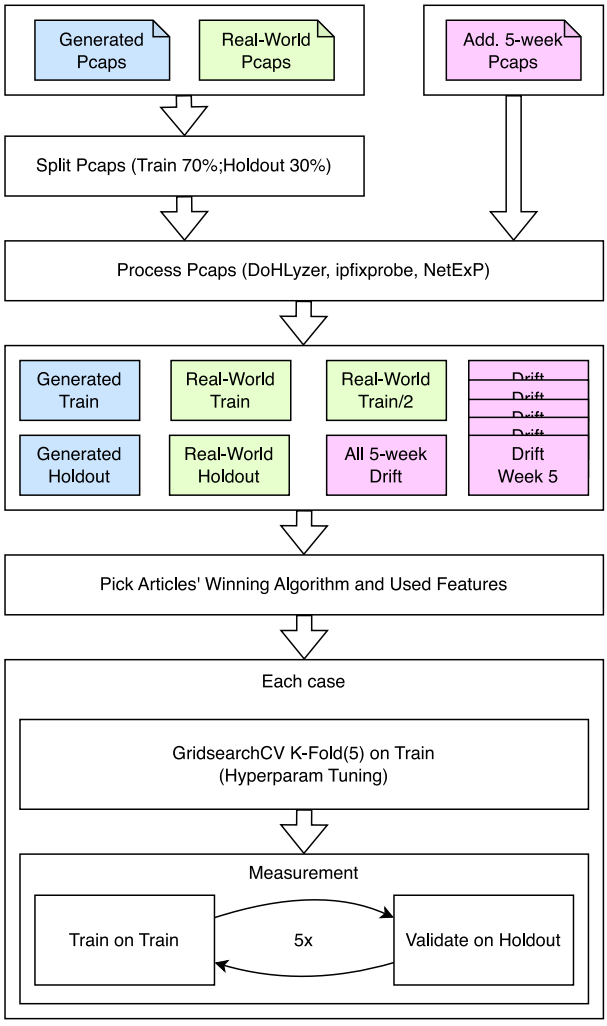

Fig. 2. Structure of general processing methodology common for all cases.

methodology was established to treat all the proposals the same way for all prepared cases. General processing methodology is depicted in Fig. 2. For the implementation, an automated pipeline was created that ensured no deviance in the methodology for each evaluated detection proposal to avoid any accidental mistakes and maintain comparable results.[4]

6.1. Data preparation

At first, the datasets described in Section 3 are in the form of raw PCAP files. Since each tool process flows differently with varying timeouts generating a nonidentical number of flows on the output, the split of the raw PCAP files into Train and Holdout parts before the flow exportation was performed. This way, it can be guaranteed that the original source is the same as the input, and the tool's processing is considered part of the study approach.

There are three groups, the Real-World, Generated, and Add. 5-week, each consisting of several PCAP files. The first two groups of files are treated differently than the third. Each file in the first two groups is split in a ratio of 70:30 to create dedicated training and holdout parts respectively. The train part is dedicated to hyper-parameters tuning of the algorithms, and then it is used to train the model for performance evaluation. The holdout part is kept exclusively for final performance evaluation. The described dataset division follows the common machine-learning methodologies and can be considered as best practice [35].

All the PCAP files are processed by each processing tool used in the selected studies (namely DoHLyzer, NetExP, and ipfixprobe) in the same way as in the original experiments. The outputs of the processing tools are then merged into several datasets that later serve in different combinations as input into different cases for answering research questions (see Fig. 3).

6.2. Algorithm selection and tuning

Only the algorithms that achieved the best classification performance in the original studies were chosen to represent each approach (see Table 3). The algorithms are fed with the appropriate features. Moreover, the recreation of the approaches includes also additional preprocessing as specified by the authors. If the published work specifies feature scaling, the same feature scaling is used. Additionally, the original works use imbalanced dataset similar to the one used in this study. During the recreation the same under-sampling, over-sampling, algorithm parameters or no action, was used the same way as specified by the authors. The algorithms were tuned using grid-search and K-Fold (5-Fold) on the train part before each measurement. The Bi-directional LSTM architecture was left intact; only hyper-parameters, such as the learning rate or optimizer algorithm that the authors also tuned, as they mentioned in their work [21], were tuned to match the used data. The best hyper-parameters were found for each training dataset (see training parts of each research question in Fig. 3). Each study model's hyper-parameters were tuned for each training dataset separately since each training dataset has different characteristics and sizes. That way, fair comparison can be achieved.

6.3. Accuracy measurement

For the final measurement, each model was trained on the whole train part using the best-found hyper-parameters for each training dataset, and then the performance was measured on the holdout part of each belonging training dataset. All the seeds were left intact (not fixed) and randomized by default at the measurement part, and for each of the five repeats, the model was retrained before measurement. Average performance and standard deviation across all five runs is then used for comparison. This way, it is possible to show more realistic results, the stability of the algorithm results, and the boundaries between which the performance can fall in reality.

$$F1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{1}$$

Since the datasets are imbalanced, the performance is measured using an F1 score defined in Eq. (1), which is a recommended measure when dealing with imbalanced data [35]. However, all methods have different constraints on the input flows. For example, Vekshin et al. [16] require flows with at least five packets; shorter flows are simply removed from the classification task, and the performance metrics are reported without them. Similarly, Montazerishatoori et al. [17] Mitshuhashi et al. [25], Behnke et al. [20], Zebin et al. [24], and Casanova et al. [21] discards flow with *NaN* values[5] and reports all the performance metric without considering them. The percentage of filtered DoH flows in each dataset is shown in Table 4. It can be noticed, that the ipfixprobe tool used by Vekshin et al. [16] discards more than 50% of all DoH flows in the Real-World dataset and directly classifies

[4] The source codes are available at https://github.com/kjerabek/comparative-analysis-of-doh.

[5] They occur due to the impossibility of feature computation—usually division by zero or short packet sequence.

Table 4
The percentage of DoH flows that were filtered by each flow-export tool due to the impossibility of feature computation. This was usually caused by a low number of packets and unidirectional communication. Such filtering results in a false-negative classification of the filtered DoH flows.

Holdout dataset	DoHLyzer	ipfixprobe	NetExP all	NetExP filter
Generated	24.3%	26.4%	32.8%	93.6%
Real-World	28.0%	50.9%	17.4%	97.1%
Add. 5-week	26.6%	26.3%	38.7%	95.8%

them as non-DoH. Moreover, NetExP Filter discards even 97.1% of all flows due to the strict filtering condition—only flows with more than 120 packets are used in classification. Not considering filtered flows in the performance metric does not reflect the true performance of the classifiers since a lot of DoH connections could be missed by such prefiltration. Therefore, two distinctive F1 metrics are used:

F1-classified is computed just from the flows that satisfied the constraints defined in the classification method. This metric was mainly used in the detection proposal studies that are replicated.

F1-all is computed from all flows in the dataset. The flows in the dataset that were filtered out since they did not satisfy the input constraints of the method are assigned with a non-DoH prediction label. This metric targets to mimic a real-deployment performance.

6.3.1. Accuracy measurement of the Jerabek et al. proposal

All selected studies in Section 5 presented only a single DoH detector proposal except the Jerabek et al. [26]. In their study, they present three main classifiers. The (1) ML-based classification of all flows (further denoted as *All Flow scenario*), the (2) ML-based classification of flows that contain more than 120 packets (further denoted as *Filtered scenario*) for improving accuracy. The minimal number of packets was selected to maintain high accuracy even with a very limited feature set. However, the filtration for a minimum of 120 packet flows is very strict; therefore, they also proposed the (3) approach, which utilizes the ML model from the Filtered scenario to create a DoH resolver blocklist. This ML-created blocklist is then used for classification, even very short flows (this approach is further denoted as a *Simulation scenario*).

Since Jerabek et al. [26] evaluate all his methods separately, all three scenarios are included in this comparative analysis. The All-flows and Filtered scenarios are directly comparable with all other approaches since they utilize ML-only. Nevertheless, the simulation scenario utilizes active probing and blocklist, giving the classifier a significant advantage in some test cases, which should be considered when interpreting the results.

6.4. Resource consumption measurement

The memory requirements of the proposals are estimated by the sizes of the models. In each performed experiment, the trained models are stored in the form of a pickle (python object dump format[6]) without any compression. The sizes of the pickled files determine the size of the models. The memory requirement statistics are then reported in the form of mean and standard deviation computed across all pickled files associated with specific proposals.

In addition, execution times of training and evaluation of the models among the inputs are measured. Since the proposed methods use different algorithms with different parallelization options and limitations, the model's training and evaluation times are measured without parallelization and use only one CPU to provide a fair comparison. Even the Bi-directional LSTM training and evaluation are limited to such constrained resources, although GPU acceleration would be used in practice.

The number of input samples varies for each model since each preprocessing tool extracts a different number of flows. Moreover, some methods are proposed to balance the input data. To combat those nuances in the number of processed samples, the detector processing speed was measured in the form of samples per second instead of using absolute training and estimation times. The metric is computed with the following formula:

$$throughput = \frac{\|X\|}{t}, \tag{2}$$

where the $\|X\|$ represents number of input samples for training or evaluation and t is the training or evaluation time. Mean and standard deviation are calculated for each method's training and evaluation runs. The measurement approach is demonstrated in Fig. 3.

The experimental setup employed a dedicated server featuring 2x AMD EPYC 7282 processors, 256 GB DDR4 RAM clocked at 3200 MHz, a 1 TB SSD, and 2x A5000 GPUs each with 24 GB RAM. This server was exclusively designated for the experiments, with no concurrent execution of other computationally intensive tasks.

7. Comparative analysis

This analysis is organized into six research questions (RQ) designed to evaluate the detectors' usability (by assessing the models' performance), transferability across diverse environments, and longevity in terms of maintaining performance over time. For each experiment, the methodology described in previous Section 6 was firmly followed. The created automated pipeline ensured no deviance in the methodology for each evaluated detection proposal to avoid any accidental mistakes and maintain comparable results. The results are presented in the following sections.

7.1. RQ1: What is the most effective DoH detection approach on a lab-created dataset?

This research question concerns the general reproducibility of the DoH detection proposals since all of them have been primarily evaluated using lab-created data. As demonstrated in Fig. 3, the pre-split Generated Train part to train the models and then the Generated Holdout to test their performance was used.

7.1.1. Results

The results of the experiment are shown in Fig. 4. The experiment confirmed the reported performance of each proposal, and the excellent accuracy was recreated, often reaching up to a mean F1-classified score of 0.99, except the Casanova et al. [21]. In this case, the experiments could not reach the expected accuracy of the classifier since they reported an F1 score of 0.987. Despite the increased effort spent in hyperparameter tuning and special care in model recreation, a slightly lower F1 score of 0.93 with a very low standard deviation of 0.001 across the runs was obtained. To validate the results, Casanova et al. were contacted; however, after two urgencies (each one month apart), no response was received.

As expected, the measured F1-all performance significantly decreases compared to F1-classified. The highest accuracy drop between the F1-all and F1-classified experienced by Jerabek et al. [26] (Filtered), who discards flows shorter than 120 packets, and Vekshin et al. [16] who discard flow with less than five payload packets.

Answer RQ1: Since almost all DoH detection approaches performed similarly well, there is no clear winner in the performance on the lab-created data. However, Casanova et al. [21], Vekshin et al. [16], and Jerabek et al. [26] in the all-flow scenario performed worse than the others. When considering F1-all measure, the most accurate are the

[6] https://docs.python.org/3/library/pickle.html.

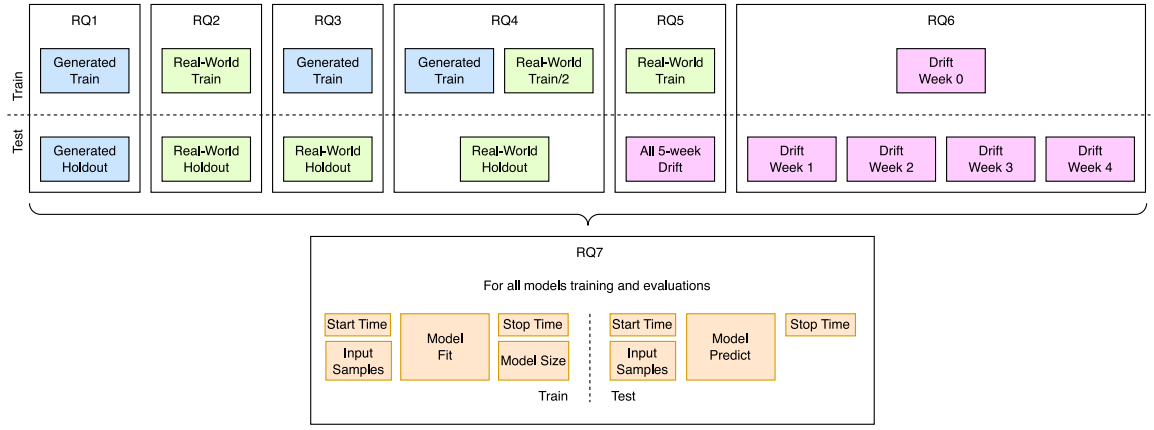

Fig. 3. Processed datasets that are used as input to answer defined research questions.

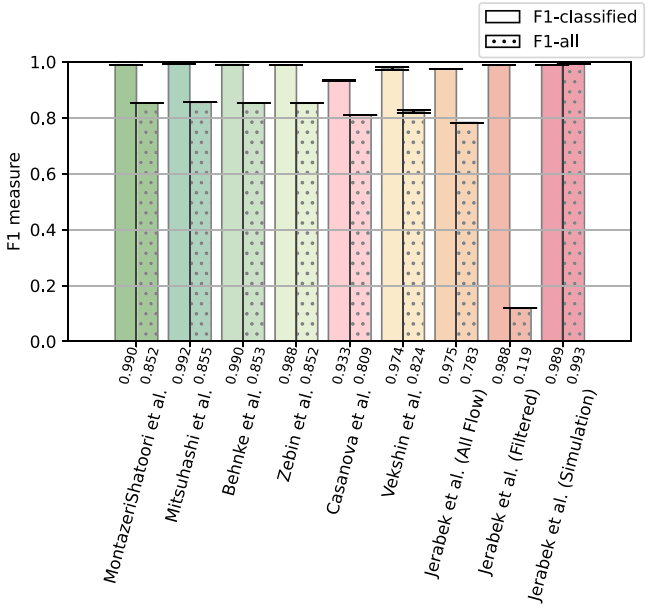

Fig. 4. RQ1: The performance of DoH detection proposals trained on the Generated dataset and tested on the Generated dataset. The exact measured values of both f1 scores are written under each column. The whiskers in the plot represent the standard deviation observed during different random seeds.

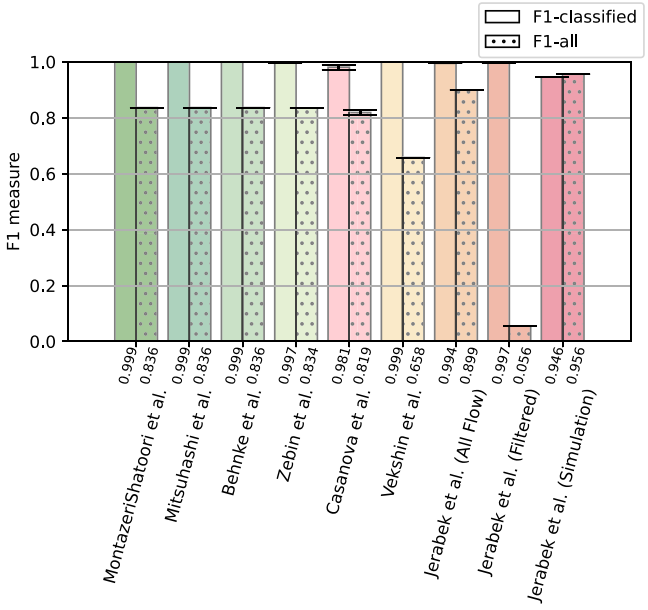

Fig. 5. RQ2: The performance of DoH detection proposals trained and tested on real-world datasets. The exact measured values of both f1 scores are written under each column. The whiskers in the plot represent the standard deviation observed during different random seeds.

approaches that use DoHLyzer as the data source, namely MontazeriShatoori [17], Mitsuhashi et al. [25], Behnke et al. [20] and Zebin et al. [24] since they perform minimal packet filtration. Additionally, Jerabek et al. [26] in the Simulation scenario also performs well, taking advantage of an accurately created blocklist.

7.2. RQ2: What is the most effective DoH detection approach on a real-world ISP dataset?

This research question aimed to validate that the approaches are similarly accurate even with real-world data. As shown in Fig. 3, a Real-World dataset for both training and testing (holdout dataset) was used.

7.2.1. Results

The results of the experiments depicted in Fig. 5 confirm that the DoH detection proposals achieve excellent performance even on the Real-World data. Since the Real-World data contain less variability in the traffic because most of the DoH is serviced by two major DoH resolvers [13], an increase in mean F1-classified scores across most of the proposals can be observed. Compared to results obtained with Generated data, Casanova et al. [21] achieved similar accuracy as reported in their study on Real-World data. Only Jerabek et al. [26] in the simulation scenario performed worse than on Generated data.

Such performance drop is caused by the lack of long connections to some DoH resolvers. Due to the condition of minimal packets in the flow (at least 120 packets), some flows are directly discarded and classified as non-DoH. Naturally, when there is no flow with at least 120 packets for some resolvers, these resolvers will always be falsely classified as non-DoH.

When looking at the performance measured with F1-all, it can be seen that by far the largest performance drop experienced by Jerabek et al. [26] in the Filtered scenario. This is caused by the shape of real-world traffic, where the majority of DoH connections are short. Similarly, Vekshin et al. [16] miss a lot of DoH connections due to deployed prefiltration to flow with at least five payload packets.

Answer RQ2: This experiment revealed that the DoH detection approaches are again very accurate in Real-World datasets. Most approaches performed better on Real-World data than on Generated data

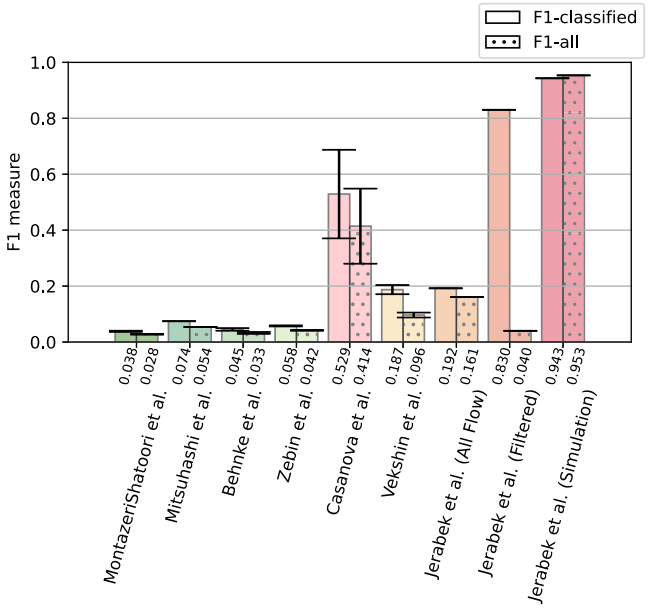

Fig. 6. RQ3: The performance of DoH detection proposals trained using Generated dataset tested on the real-world dataset. The exact measured values of both f1 scores are written under each column. The whiskers in the plot represent the standard deviation observed during different random seeds.

(in RQ1), except the Casanova et al. [21], and Jerabek et al. [26] (in both Simulated and all-flows scenario), where they achieved slightly lower mean F1-classified and F1-all score than the others. Similarly, as in the previous RQ1, when concerning F1-all, the most accurate approach uses the DoHLyzer as the data source.

7.3. RQ3: How effective are the lab-created DoH detectors in the real-world ISP environment?

This research question aims to evaluate the usability of the detectors trained on the laboratory-generated dataset in a production environment. This experiment supports cases where it is not always possible to collect the data from the final deployment environment due to various restrictions, e.g., to preserve privacy. Additionally, it may be hardly achievable to label all the captured encrypted connections correctly—for example, due to incomplete IP or domain labeling lists. Finally, this experiment also assesses the robustness of the models and their dependability on the network environment used during training. Ideally, the models should not be dependable on training network environments and should be applicable in diverse deployment settings. To simulate this deployment setup, the trained part of the Generated dataset for training was used, and all detectors used the holdout part from the Real-World dataset.

7.3.1. Results

The results are shown in Fig. 6. As can be seen, the models trained using the Generated dataset are mostly struggling to classify DoH accurately, and most of them are unusable. The only approach that remains usable is the Jerabek et al. [26] in filtering and simulation scenarios as it achieves almost unchanged F1-classified performance as in the previous case. The reason behind such a difference can be attributed to the strict filtering condition of at least 120 packets, where the DoH traffic shape became very distinctive. The traffic shape of shorter DoH flows is highly influenced by the TCP/TLS handshakes or HTTP/2 preface, and it is difficult to recognize DoH from other short HTTPs communication [26,31]; since the datasets come from a different environment, the classifiers fail to generalize on short flows and misclassify them.

The approach that also achieved far better performance than the others is the Casanova et al. [21] despite its lowered performance on Generated holdout. The reasons behind the increased performance are unknown due to the low explainability of the used neural networks.

Answer RQ3: As shown in the results, most of the plain ML-based methods are ineffective when trained in one environment and deployed in another. According to the original data article [13] the Generated dataset contains traffic from more resolvers than the Real-World dataset—the Generated dataset contains a wider variety of different DoH implementations, yet the detectors in the real world did not perform well. Therefore, it can be assumed, that the inapplicability of the Generated dataset is caused by an entirely different environment between the lab and real networks.

The used Generated dataset is the most comprehensive lab-created dataset available since it contains traffic from 16 different resolvers. The other known publicly available datasets contain far fewer traffic traces to a smaller number of resolvers (only 4 in case of CIRA-CIC-DoHBrw-2020 [7], only 2 in case of dataset [9] used by Vekshin et al. [16]). Therefore, the observation can be considered as generally applicable, and the lab-created dataset must be used only as a benchmark and not in production. The deployment into the real-world environment is not concerned by the DoH detector proposals, except the Jerabek et al. [26], who also performed the best in this experiment with their Simulation scenario. In conclusion, most of the DoH detector proposals are not robust, and their high accuracy depends on the similarities between the training and deployment networks.

7.4. RQ4: How effective are DoH detectors trained on both types of datasets (lab-created and real-world mixed together) in detecting DoH traffic in real-world ISP deployment?

The research question #4 aims to find out the influence of the Generated dataset on real-world performance. The Real-World dataset contains mainly two DoH resolvers [13]. The data are similar to the real-world environment but lack the variability of the generated dataset. Therefore, the rare DoH resolvers might get misclassified as non-DoH.

In this experiment, the training parts from the Real World and Generated dataset were mixed. The whole Generated training dataset and 1/2 of the Real-World training dataset were used as graphically demonstrated in Fig. 3.

From the deployment perspective, this scenario can be seen from two angles. The first covers the case where only a laboratory-created dataset is available, and it is planned to deploy the detector in the real network. Hence it is beneficial to add some traffic from the real network to the training set. On the other side, it could also represent a case where only a limited amount of data from a real network is available; hence laboratory data (data from another environment) are added to provide more rich behavior and make the model more robust.

7.4.1. Results

The results of the performed experiments are shown in the in Fig. 7. As can be seen, the performance of the approaches experienced a small decrease in performance compared to the training on the Real-World data only shown in Fig. 5. The biggest decrease was experienced by Jerabek et al. [26] in the all-flow scenario. The drop can be attributed to the use of only four features, which limits the capability of covering the nuances of the more complex dataset. However, the simulation scenario is stable compared to training on only Real-World data and performs similarly. Moreover, Casanova et al. [21] Bi-directional LSTM approach showed increased and more stable performance, which is now comparable to the other methods with the DoHLyzer data source.

Answer RQ4: Experiments showed that the DoH approaches are still very effective. However, the mixture of the Real-World with the Generated dataset does not bring many benefits but rather slightly decreases the performance of most detectors compared to the case when trained on the Real-World dataset only.

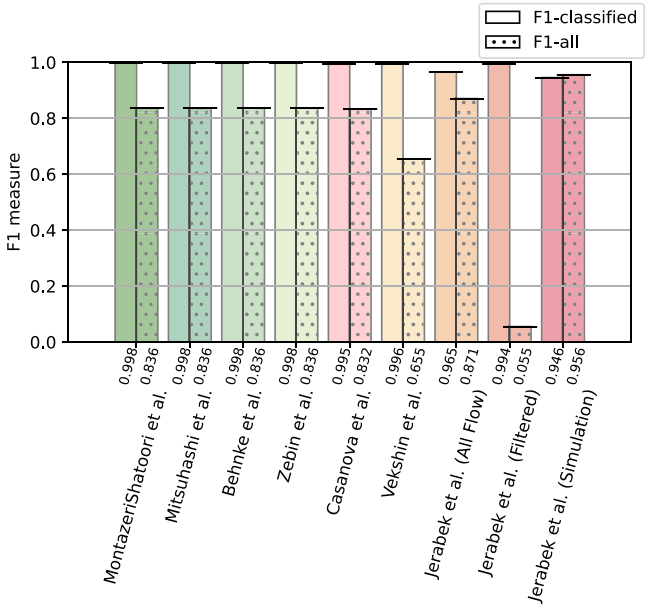

Fig. 7. RQ4: The performance of DoH detection proposals trained using Generated dataset and 1/2 of real-world train dataset. Proposals were tested on the real-world holdout dataset. The exact measured values of both f1 scores are written under each column. The whiskers in the plot represent the standard deviation observed during different random seeds.

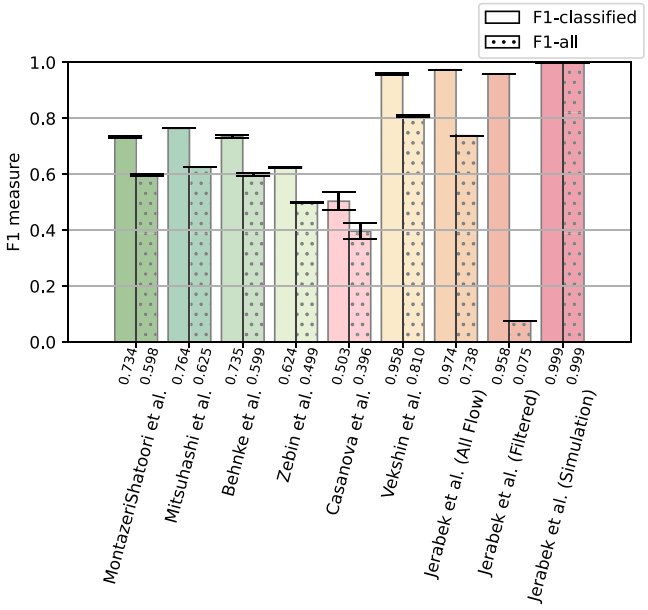

Fig. 8. RQ5: The performance of proposed detectors trained on the real-world dataset and evaluated on the whole additional 5-week dataset. The time difference between the training and testing dataset is at least one year.

7.5. RQ5: Are concept or data drifts a significant phenomenon in the DoH detection task?

Drifts (either data or concept) are an important phenomenon that needs to be considered when designing the network detector. The longevity of detectors is an important parameter that depends on various factors, such as the detector design, underlying network infrastructure, or specific time in a year that causes different network traffic shapes or simply changes in the classified service itself.

In this research question #5, the susceptibility of DoH detectors to drifts on the CESNET2 network backbone environment was tested. It can be confirmed, that during the time, there was no change in the CESNET2 network or monitoring infrastructure; thus, the drift originates from the traffic itself. The extreme case was evaluated, where the training dataset was captured at least one year before the evaluation data to test the longevity of the detectors. Therefore, the train part of the Real-World dataset was used for training, and the Additional 5-week dataset was used for evaluation.

7.5.1. Results

The results depicted in Fig. 8 show an expected significant performance drop across all the approaches except the Jerabek et al. [26] and Vekshin et al. [16] that proved to be more resistant to the longitudinal drift and suffered only small or no performance degradation. Moreover, Jerabek et al. [26] in the simulation case very effectively mitigated the drift with his adaptive blocklist approach.

Compared to Jerabek et al. [26] and Vekshin et al. [16], the approaches using DoHLyzer as data source proved to be more affected by the longitudinal drifts in both F1-classified and F1-all measures.

Answer RQ5: Despite the extreme case of training and evaluation data captured more than a year apart, some of the approaches still maintain their high accuracy—Vekshin et al. [16] and Jerabek et al. [26] achieved F1-classified score over 0.95, which is similar to the score reported in their proposing studies. Both approaches proposed the omitting of the first connection packets from the feature vectors. These packets carry mainly TLS and HTTP-dependent information and are unrelated to the actual DoH data. Removing these packets significantly improved longevity since other approaches that include all packets in the feature vector did not (by far) perform significantly worse in the experiment.

7.6. RQ6: How DoH detection performance degrade over a month in real-world ISP networks?

The previous research question showed the extreme case of long drift. The research question RQ6 aim to examine the drift over a single month. For this experiment, only the Additional 5-week dataset was used. As shown in Fig. 3, the training and hyperparameter tuning was performed on the first week, the same way as described in Section 6, while the detectors were evaluated on the remaining weeks separately to observe the potential degradation in the classification performance. In the case of the simulation scenario of Jerabek et al. [26], the blocklist was reset so that the evaluation for each weak starts with the empty blocklist.

7.6.1. Results

The per-week accuracy of each approach is shown in Fig. 9. A common trend in performance changes can be noticed. The high performance in Week 2, which then decreases in Week 3. In Week 4, there is a small increase, followed by a more noticeable decrease in Week 5. The last large decrease is common to all measured approaches and can be attributed to Christmas. Since the CESNET2 network is mainly used by universities and research institutions in the Czech Republic, a public holiday such as Christmas often results in significant traffic shape and distribution changes due to a lower amount of transferred data.

Most of the approaches performed with similar accuracy as reported in the original studies and achieved F1-classified scores over 0.99, with the exception of Vekshin et al. [16] and Casanova et al. [21].

The performance of Vekshin et al. [16] significantly drops in Week 3 and Week 5. This performance instability can be attributed to the smaller training set compared to previous experiments. The smaller set was probably overfitted, and the model then struggled to deal with the network changes.

Casanova et al. [21], as in previous cases, poses a large variance in the performance depending on the run. The F1-classified score is similar

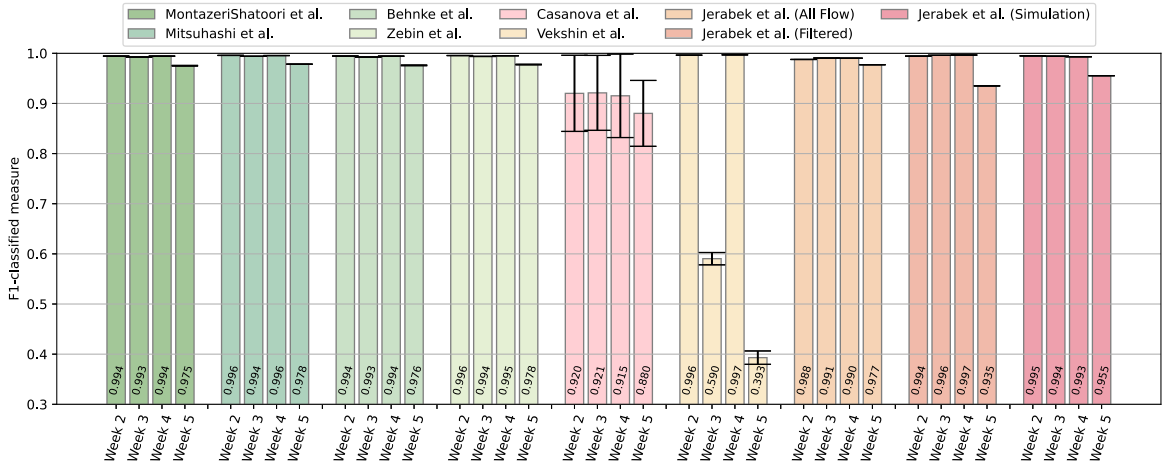

Fig. 9. RQ6: Per-week F1-classified performance of the DoH detectors trained on week 1 of the additional 5-week dataset. The whiskers in the plot represent the standard deviation observed during different random seeds.

to the one achieved on the Generated dataset, which is ~0.05 lower than reported in the original study.

Answer RQ6: The experiments show that drift in the short term does not significantly affect the performance of the classifiers, except for Vekshin et al. [16] and Casanova et al. [21].

7.7. RQ7: What are the resource requirements of the proposed detectors?

Another critical aspect besides the achieved accuracy of the proposed methods to be considered for practical usage is the resource consumption of the ML-based detectors. High throughput and low complexity determine the applicability of models in real networks. Hence, estimating the size and complexity of the trained models is essential.

In addition to the model size, model throughput shows the amount of flows processed by the model, determining their computational effectiveness. Moreover, models would be periodically retrained as the underlying data changes over time. The speed of training is another parameter that needs to be considered.

7.7.1. Results

The training performance of the models is depicted in Fig. 10. It can be seen that there is a high disparity between the training throughput of the models, which can be caused by the use of different ML algorithms and their particular settings. The highest throughput of samples during the training is achieved in the case of Mitsuhashi et al. [25] and Jerabek et al. [26] in both All Flows and Filtered (Simulation is not included as it uses the same model as in the Filtered scenario). Both Jerabek et al. [26] and Mitsuhashi et al. [25] use XGBoost classifier[7] and their speed difference compared to others is significant. Nevertheless, Jerabek et al. [26] achieved the fasted training speeds in both scenarios since their classifier is trained on only four features.

The other approaches are much slower. MontazeriShatoori et al. [17] and Behnke et al. [20] achieved similar training speeds, since both are using Random Forest Classifier. Vekshin et al. [16] utilizes Ada-Boost which performs close to the previous two. Nevertheless, the Ada-Boost cannot be efficiently parallelized; thus the gap between Random Forests would be much higher in parallelized environments. Casanova et al. [21] is next in the row. It is the only approach utilizing neural networks. The slowest is the Zebin et al. [24] approach that uses a stacking classifier.

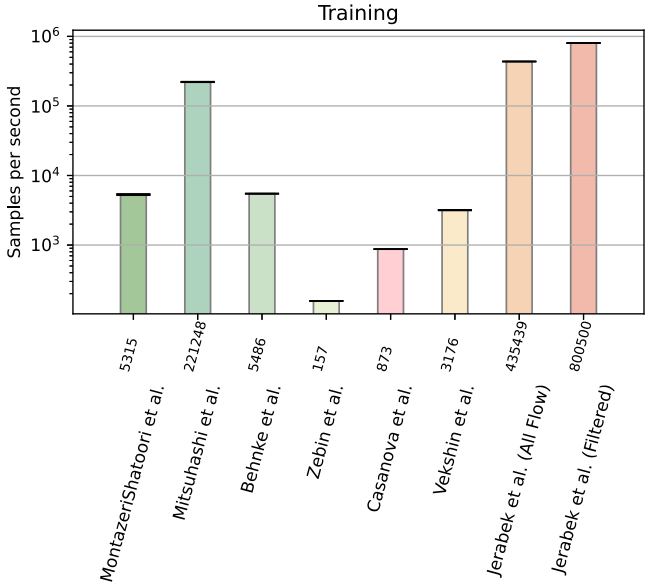

Fig. 10. Training throughput samples per second of the models during the training in log scale.

The prediction throughput of all the trained models was measured, and the results can be seen in Fig. 11. The throughput of most models reaches more than 400K samples per second except for Zebin et al. [24] stacking classifier approach and Casanova et al. [21] Bi-directional LSTM approach. The two mentioned may suffer from the increased complexity of the used models. Higher throughput is achieved by Vekshin et al. [16] with Ada-Boost classifier. Their classifier may benefit from being built with an ensemble of only 50 estimators, while other ensemble models were built with a number of estimators equal to 100. The highest detection throughput is achieved by the ML model of Jerabek et al. [26] in a filtered scenario since it is lightweight and simple. However, this advantage comes at the cost of a reduced F1-all accuracy.

Finally, the memory consumption of models is measured among all trained models in previous RQs, and the mean and standard deviation are provided in Table 5. It can be seen that some models are reaching up to tens or even hundreds of MBs. The stacking classifier proposed by Zebin et al. [24] is the most complex and reaches the largest sizes. Vekshin et al. [16] Ada-Boost occupies lower sizes. The most efficient

[7] Python XGBoost library https://xgboost.readthedocs.io/.

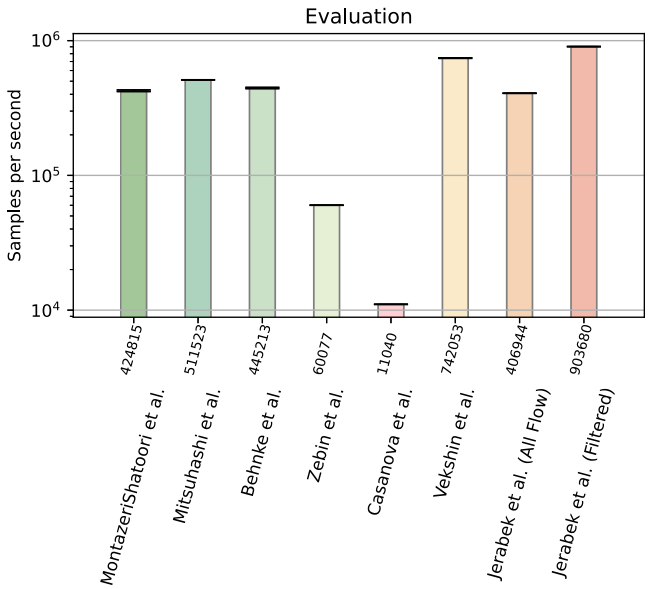

Fig. 11. Evaluation throughput samples per second of the models during prediction in log scale.

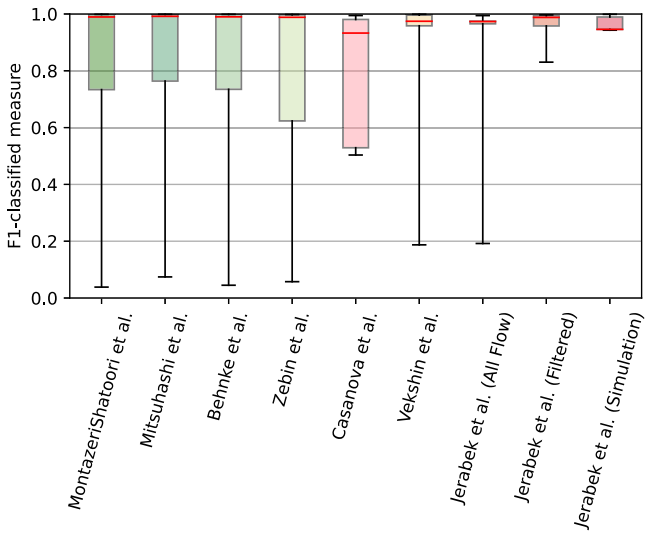

Fig. 12. Boxplot of the measured mean F1-classified performances for each DoH detection proposal across experiments performed in RQ1–RQ5. The whiskers represent the minimum or maximum observed value.

Table 5
Model sizes mean among all measurement cases.

Method	Mean [MB]	STD
MontazeriShatoori et al. [17]	170.54	108.16
Mitsuhashi et al. [25]	5.17	3.54
Behnke et al. [20]	165.47	105.64
Zebin et al. [24]	1001.83	677.23
Casanova et al. [21]	0.19	0.00
Vekshin et al. [16]	31.76	19.63
Jerabek et al. [26] (All Flow)	9.13	6.28
Jerabek et al. [26] (Filtered)	0.54	0.30

memory consumption are the XGBoost models proposed by Jerabek et al. [26], Mitsuhashi et al. [25] and Casanova et al. [21]. The neural network of Casanova et al. [21] required constant size as it stores mainly weights and it remained unchanged during the experiments.

Answer RQ7: Experiments have shown that resource consumption and effectiveness vary highly amongst the approaches. The most effective in training, evaluation, and memory consumption were the approaches utilized XGBoost Classifier. While more complex models such as Zebin et al. [24] show increased resource requirements without an actual increase in accuracy.

In this RQ, the models were measured using only one CPU. In reality, parallelization should be possible. Not all algorithms can be parallelized equally well or at all during the model training. Hence, this study limits the measurement to equal conditions to make the results comparable. However, the models' evaluation throughput would scale linearly when run in parallel.

Last but not least, the measured metrics of the models are strongly dependent on the used machine learning library. Proposals based on scikit-learn library usually showed worse metrics compared to the xgboost-based detectors. Nevertheless, since the studies were recreated with the same Python libraries as in the original works; thus, the performance results are consistent with the proposals.

8. Discussion

In this study the best-performing ML-based detectors were evaluated in different scenarios. It can be noticed that the best performing across the DoH detection proposals are usually a tree-based algorithms, with the exception of Casanova et al. [21], who evaluated only neural networks. However, the neural network showed a large variability in performance. Fig. 12 depicts the overall F1-classified performance across all the evaluated cases. Jerabek et al. [26] and Vekshin et al. [16] benefited from the additional filtration, and their performance showed a lower variance across the experiments. The other approaches based on the feature provided by DoHLyzer show similar performance and variance. Nevertheless, the detector proposed by Mitsuhashi et al. [25] achieved a slightly better performance than other DoHLyzer-based classifiers.

The results showed that the lab-created datasets can be far more challenging than the real-world ones. The DoH detectors achieved better performance when trained and tested on the Real-World dataset compared to the Generated one. Nevertheless, most of the ML models proved to be unusable in real-world deployment when trained on the lab-created dataset or generally when trained on the data from one particular computer network and deployed in a different one.

Performance of the models in the RQ3 experiment (trained on the Generated and evaluated on Real-World datasets) differed vastly. Detectors that performed better in RQ1 and RQ2 (trained and evaluated on similar network data) failed when used with data from different network environments. More simple but generally less accurate models (in order of 0.01 of F1 score) showed higher transferability, and they can handle really long time between training and deployment due to higher generalization.

The drift (caused by the old training set) can be considered a relatively small problem in the case of DoH detection. Even one year after training, detectors showed relatively good performance (RQ5 experiment). Nevertheless, regular model retraining on at least a monthly basis is still recommended to maintain stable and consistent accuracy results over time. Besides, the models should be trained on a sufficient amount of data, to mitigate the effect observed with Vekshin et al. [16], which failed to generalize and showed a significant performance variability when trained on a single week data.

Overall, the DoH detection problem can be considered as almost solved when the ML-based detector is deployed together with the blocklist—as proposed by Jerabek et al. [26]. The block-list-based solution showed superior performance over ML-only proposals and can detect even single-query flows, which are considered a major challenge by the Hynek et al. [6] survey. However, small and private DoH resolvers used stealthily (with very short DoH connections, e.g., for

Command & Control server access) would still be missed even with current solutions.

The blocklist-based solution can be combined with any of the evaluated ML models since all of them showed high performance. The selection depends on the deployment factors—available flow data source, network infrastructure, and the possibility of retraining. When frequent retraining is not possible (e.g., due to privacy policies preventing frequent and automatic traffic captures), Jerabek et al. [26] or Vekshin et al. [16] could be considered reasonable candidates due to their longitudinal stability. From these two approaches, the Jerabek et al. [26] has a minimal feature vector that is lightweight to compute and can be obtained from any flow monitoring device (including switches and firewalls), which makes their approach far more deployable compared to Vekshin et al. [16].

On the other hand, when retraining is possible, Mitshuhashi et al. [25] achieved slightly better performance compared to other detectors that tend to overfit. Nevertheless, according to Jerabek et al. [26], their features cannot be extracted efficiently in high-speed networks. Therefore, the deployment of the detector proposed by Mitshuhashi et al. [25] is also limited to slower networks only. In addition, the three mentioned approaches also outperforms the other approaches in the effectiveness of speed and resource consumption.

9. Conclusion

DoH is a privacy technology that has already gained adoption among users and also service providers. One of the privacy-preserving features is its stealthiness since DoH blends into other HTTPS traffic, leaving the administrators unaware of its presence. Therefore, multiple DoH detection approaches have been proposed in recent years to notify security personnel about ongoing DoH communication. In this study, a comparative analysis of the most influential flow-based DoH proposals was performed to evaluate their performance independently and summarize the state and deployability of current solutions.

This study covers seven previously published recreated detectors and evaluated them in six different experiments to answer defined research questions. Each research question targeted specific deployment scenarios and concerned ML-model transferability, usability, and longevity. The experimental methodology has been settled in advance and automated to avoid any accidental mistakes and maintain the trustworthiness of the results. Moreover, the largest collection of DoH datasets was used, which enabled testing the classifiers in real-world and laboratory environments. A novel Additional 5-week dataset [29] was also created, which enabled experiments concerning the longevity of the detectors.

Presented results confirmed the reported high accuracy of DoH detection proposals, achieving an F1 score of 0.99. However, the experiments also showed that the transferability of detectors to different network environments and their differences in the rate of obsolescence pose a challenge that the research community should focus on in the future.

Overall, this study showed that traditional detection of benign DoH can be considered a practically deployable solution when used in the simulated scenario presented by Jerabek et al. [26]. The approach of detection blocklists created by machine learning showed superior performance and was even transferable between network environments. Moreover, it can be combined with any other ML-based detector, depending on the deployment constraints. On the other hand, malicious users that use DoH stealthily (e.g., by using only short connections) can still bypass the current DoH detection proposals, and researchers should focus more on early DoH detection from the first few packets sent at the beginning of the connections.

The study validated several scenarios including deployment of models trained in one network to another. However, the presented case covers only two networks with certain conditions. In future work, the authors would like to explore the models transferability to networks in different geographical locations with varying conditions such as different bandwidth, delay, jitter or packet loss.

Ethical statement

The privacy of the CESNET2 network users was the highest priority during the creation of an Additional five-week dataset; therefore the research has been done with extreme carefulness. The indisputable advantages of real traffic generated by hundreds of thousands of people come with the cost of potential privacy abuse of real users. Therefore, only automatic data processing was used with immediate data anonymization. With previous statements in mind, there was no manual analysis of deanonymized data, and no procedures that could reveal real user identity were performed.

CRediT authorship contribution statement

Kamil Jerabek: Writing – original draft, Validation, Software, Methodology, Investigation, Data curation, Conceptualization. **Karel Hynek:** Writing – original draft, Validation, Software, Methodology, Investigation, Conceptualization. **Ondrej Rysavy:** Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

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

Data availability

The data used for this research are available at the Zenodo platform. The link can be found in the article references.

Acknowledgments

This research was funded by: the Ministry of Interior of the Czech Republic, grant No. VJ02010024: Flow-Based Encrypted Traffic Analysis; Brno University of Technology, Faculty of Information Technology internal grant FIT-S-23-8209.

References

[1] P.E. Hoffman, P. McManus, DNS Queries over HTTPS (DoH), Technical Report 8484, Internet Engineering Task Force, 2018, http://dx.doi.org/10.17487/RFC8484.

[2] S. García, J. Bogado, K. Hynek, D. Vekshin, T. Čejka, A. Wasicek, Large scale analysis of DoH deployment on the internet, in: V. Atluri, R. Di Pietro, C.D. Jensen, W. Meng (Eds.), Computer Security – ESORICS 2022, Springer Nature Switzerland, Cham, 2022, pp. 145–165.

[3] S. Deckelmann, Firefox continues push to bring DNS over HTTPS by default for US users, 2020, URL: https://blog.mozilla.org/blog/2020/02/25/firefox-continues-push-to-bring-dns-over-https-by-default-for-us-users/. (Accessed May 2021).

[4] M. Huc, How to enable DNS over HTTPS (DoH) on Windows 11, 2022, URL: https://pureinfotech.com/enable-dns-over-https-windows-11/. (Accessed December 2022).

[5] Quad9 Foundation, iOS and MacOS mobile provisioning profiles are here!, 2022, URL: https://www.quad9.net/news/blog/ios-mobile-provisioning-profiles/. (Accessed December 2022).

[6] K. Hynek, D. Vekshin, J. Luxemburk, T. Cejka, A. Wasicek, Summary of DNS over HTTPS abuse, IEEE Access 10 (2022) 54668–54680, http://dx.doi.org/10.1109/ACCESS.2022.3175497.

[7] M. MontazeriShatoori, L. Davidson, G. Kaur, A.H. Lashkari, CIRA-CIC-DoHBrw-2020, 2020, URL: https://www.unb.ca/cic/datasets/dohbrw-2020.html. (Accessed May 2022).

[8] K. Jeřábek, S. Stuchlý, DNS over HTTPS network traffic, 2021, http://dx.doi.org/10.21227/96ea-2055.

[9] D. Vekshin, K. Hynek, T. Cejka, Dataset used for detecting DNS over HTTPS by machine learning, 2020, http://dx.doi.org/10.5281/zenodo.3906526.

[10] I.F. Kilincer, F. Ertam, A. Sengur, Machine learning methods for cyber security intrusion detection: Datasets and comparative study, Comput. Netw. 188 (2021) 107840, http://dx.doi.org/10.1016/j.comnet.2021.107840, URL: https://www.sciencedirect.com/science/article/pii/S1389128621000141.

[11] M. Zwilling, G. Klien, D. Lesjak, Ł. Wiechetek, F. Cetin, H.N. Basim, Cyber security awareness, knowledge and behavior: A comparative study, J. Comput. Inf. Syst. 62 (1) (2022) 82–97, http://dx.doi.org/10.1080/08874417.2020.1712269.
[12] A. Shahraki, M. Abbasi, A. Taherkordi, A.D. Jurcut, A comparative study on online machine learning techniques for network traffic streams analysis, Comput. Netw. 207 (2022) 108836.
[13] K. Jeřábek, K. Hynek, T. Čejka, O. Ryšavý, Collection of datasets with DNS over HTTPS traffic, Data Brief (2022) 108310.
[14] K. Bumanglag, H. Kettani, On the impact of DNS over HTTPS paradigm on cyber systems, in: 2020 3rd International Conference on Information and Computer Technologies, ICICT, 2020, pp. 494–499, http://dx.doi.org/10.1109/ICICT50521.2020.00085.
[15] J. Bushart, C. Rossow, Padding ain't enough: Assessing the privacy guarantees of encrypted {DNS}, in: 10th USENIX Workshop on Free and Open Communications on the Internet (FOCI 20), 2020.
[16] D. Vekshin, K. Hynek, T. Cejka, Doh insight: Detecting dns over https by machine learning, in: Proceedings of the 15th International Conference on Availability, Reliability and Security, 2020, pp. 1–8.
[17] M. MontazeriShatoori, L. Davidson, G. Kaur, A.H. Lashkari, Detection of doh tunnels using time-series classification of encrypted traffic, in: 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), IEEE, 2020, pp. 63–70.
[18] K. Hynek, T. Cejka, Privacy illusion: Beware of unpadded DoH, in: 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON, 2020, pp. 0621–0628, http://dx.doi.org/10.1109/IEMCON51383.2020.9284864.
[19] Y.M. Banadaki, Detecting malicious dns over https traffic in domain name system using machine learning classifiers, J. Comput. Sci. Appl. 8 (2) (2020) 46–55.
[20] M. Behnke, N. Briner, D. Cullen, K. Schwerdtfeger, J. Warren, R. Basnet, T. Doleck, Feature engineering and machine learning model comparison for malicious activity detection in the dns-over-https protocol, IEEE Access 9 (2021) 129902–129916.
[21] L.F.G. Casanova, P.-C. Lin, Generalized classification of DNS over HTTPS traffic with deep learning, in: 2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), IEEE, 2021, pp. 1903–1907.
[22] H. Jha, I. Patel, G. Li, A.K. Cherukuri, S. Thaseen, Detection of tunneling in DNS over HTTPS, in: 2021 7th International Conference on Signal Processing and Communication, ICSC, IEEE, 2021, pp. 42–47.
[23] J. Wu, Y. Zhu, B. Li, Q. Liu, B. Fang, Peek inside the encrypted world: Autoencoder-based detection of doh resolvers, in: 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE, 2021, pp. 783–790.
[24] T. Zebin, S. Rezvy, Y. Luo, An explainable AI-based intrusion detection system for DNS over HTTPS (DoH) attacks, IEEE Trans. Inf. Forensics Secur. 17 (2022) 2339–2349, http://dx.doi.org/10.1109/TIFS.2022.3183390.
[25] R. Mitsuhashi, Y. Jin, K. Iida, T. Shinagawa, Y. Takai, Detection of DGA-based malware communications from DoH traffic using machine learning analysis, in: 2023 IEEE 20th Consumer Communications & Networking Conference, CCNC, 2023, pp. 224–229, http://dx.doi.org/10.1109/CCNC51644.2023.10059835.
[26] K. Jerabek, K. Hynek, O. Rysavy, I. Burgetova, DNS over HTTPS detection using standard flow telemetry, IEEE Access 11 (2023) 50000–50012, http://dx.doi.org/10.1109/ACCESS.2023.3275744.
[27] M. Lyu, H.H. Gharakheili, V. Sivaraman, A survey on DNS encryption: Current development, malware misuse, and inference techniques, ACM Comput. Surv. 55 (8) (2022) http://dx.doi.org/10.1145/3547331.
[28] N. Malekghaini, E. Akbari, M.A. Salahuddin, N. Limam, R. Boutaba, B. Mathieu, S. Moteau, S. Tuffin, Deep learning for encrypted traffic classification in the face of data drift: An empirical study, Comput. Netw. 225 (2023) 109648, http://dx.doi.org/10.1016/j.comnet.2023.109648, URL: https://www.sciencedirect.com/science/article/pii/S1389128623000932.
[29] K. Jeřábek, K. Hynek, O. Ryšavý, Five-week DoH dataset collected on ISP backbone lines, 2023, URL: https://doi.org/10.5281/zenodo.8348773.
[30] P. Mockapetris, Domain Names - Implementation and Specification, Technical Report 1035, Internet Engineering Task Force, 1987, http://dx.doi.org/10.17487/RFC1035, URL: https://www.rfc-editor.org/info/rfc1035.
[31] K. Jerabek, O. Rysavy, I. Burgetova, Measurement and characterization of DNS over HTTPS traffic, 2022, URL: https://doir.org/10.48550/ARXIV.2204.03975.
[32] M. Konopa, J. Fesl, J. Jelínek, M. Feslová, J. Cehák, J. Janeček, F. Drdák, Using machine learning for DNS over HTTPS detection, in: Proceedings of 19th European Conference on Cyber Warfare and Security, 2020, p. 205.
[33] T.A. Nguyen, M. Park, Doh tunneling detection system for enterprise network using deep learning technique, Appl. Sci. 12 (5) (2022) 2416, http://dx.doi.org/10.3390/app12052416.
[34] L.F.G. Casanova, P.-C. Lin, et al., Malicious network traffic detection for DNS over HTTPS using machine learning algorithms, APSIPA Trans. Signal Inf. Process. 12 (2) (2023).
[35] J. Han, J. Pei, H. Tong, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2022.

Kamil Jerabek is a Ph.D. candidate at Faculty of Information Technology, Brno University of Technology. He started his Ph.D. studies in 2017. He has participated in several research projects, either national or international. His research interest is focused on topics related to computer networks, cybersecurity, big data, data analysis, and machine learning. He is part of the Network and Distributed Systems Research Group.

Karel Hynek is currently working in Research and Development at the CESNET association. He received his Ph.D. study in 2023 with a topic related to network security and network monitoring. Since his bachelor's study, Ing. Hynek has worked on several national and international research projects related to security, and he is currently a leader and key member of the network traffic monitoring research team.

Ondrej Rysavy received the Ph.D. degree in computer science from the Brno University of Technology, Brno, Czech Republic, in 2005. He is an Associate Professor with the Department of Information Systems, Brno University of Technology, Brno. His research interests include computer networking and, in particular, network monitoring, network security and forensics, and network architectures. His work is focused on improving network security through data analysis by application of data mining, statistics, and distributed computing.