IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON DATA MINING FOR INTERNET OF THINGS

Received January 2, 2020, accepted January 14, 2020, date of publication January 23, 2020, date of current version January 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2969015

Traffic Fingerprinting Attacks on Internet of Things Using Machine Learning

MONIKA SKOWRON[1], ARTUR JANICKI[1], (Member, IEEE), AND WOJCIECH MAZURCZYK[2], (Senior Member, IEEE)

[1]Institute of Telecommunications, Warsaw University of Technology, 00-661 Warsaw, Poland
[2]Institute of Computer Science, Warsaw University of Technology, 00-661 Warsaw, Poland

Corresponding author: Wojciech Mazurczyk (wmazurcz@ii.pw.edu.pl)

This work was supported, under the Secure Intelligent Methods for Advanced Recognition of Malware and Stegomalware (SIMARGL) project by the European Commission under Horizon 2020 research and innovation programme under Grant 833042.

ABSTRACT The Internet of Things (IoT) concept has been widely adopted and Internet connected devices enter more and more areas of our everyday lives. However, their limited security measures raise increasing concerns, especially in terms of users' privacy. That is why, in this paper, privacy risks, focusing primarily on information leakage exposed by traffic fingerprinting attacks, on IoT devices are investigated. The considered attacks take advantage of the statistical network flows' features and application of machine learning (ML) methods and can be utilized by a passive traffic observer. In this perspective, the first part of the research presented in this paper analyzes the feasibility of identifying individual devices in a victim's home network. It considers smart environment setups of different scales and conditions, and it also includes a performance comparison of the different ML models applied. The second part introduces and validates a method for the devices' state detection based on pattern recognition with ML. Finally, recommendations for mitigating the discussed privacy risks are also enclosed.

INDEX TERMS Internet of Things, machine learning, network traffic fingerprinting, privacy, traffic analysis.

I. INTRODUCTION

The concept of everyday objects connected to the Internet is no longer a visionary idea, but a phenomenon already widespread and adopted in today's world. The term Internet of Things (IoT), which was first introduced by Kevin Ashton in 2009 [1], has evolved rapidly together with the growth of Internet technology and is now highly commercialized. As the Gartner agency predicts, the number of connected devices will reach more than 20 billion in 2020, with most of them for consumer use.[1]

Common benefits that the IoT brings is automation of processes and monitoring as well as increased intuitiveness of the environment by providing a combination of sensing, communication and computing services (among others), and enabling access to these services on demand [2]; due to these, the IoT has reached a vast variety of applications. The most popular uses for the IoT include[2] wearables, smart home systems, smart grids, smart cities, smart-parking solutions, waste management, connected cars and industrial IoT, etc.

However, the rapid development of the IoT concept does not go together with proper adoption of security measures, which might put a user and their privacy in danger. User privacy is often referred to as one of the most critical security concerns regarding IoT solutions. Indeed, IoT devices tend to store sensitive or personally identifiable information about the user, such as names, phone numbers, addresses, behavior patterns and habits. From this perspective, sometimes inappropriate measures are taken to store, process and transport this data in a secure way. The common vulnerabilities of IoT devices allow adversaries to take control over them or steal private information [3], [4]. However, user privacy can not only be threatened by an active attacker, who wants to compromise these devices; an improperly protected communication layer of the IoT solutions create an opportunity for a passive eavesdropper to extract sensitive information about

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

[1]https://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016

[2]https://iot-analytics.com/10-internet-of-things-applications

the user from the network traffic. Often it is possible because the messages sent by devices are not encrypted. However, even if encryption was in place, characteristics of the traffic, such as packet sizes and traffic rates, may expose the user's current activities.

One of the network-related attacks on users' privacy that takes advantage of these characteristics is traffic fingerprinting [5]. This method has been already successfully used for inferring what websites a user is browsing, which applications or devices they are using as well as for learning their personality traits, beliefs, health condition or personal opinions through their online behavior [6]. Yet the potential of traffic fingerprinting in IoT has drawn the attention of researchers only in recent years and still leaves many open questions [7]–[9].

In this article we investigate the risks related to IoT traffic analysis. We propose a 2-stage classification for device identification and, afterwards, recognition of their states (which are closely related to the user's actions). The introduced solution is evaluated using two datasets: one, that consists of self-collected packet traces from real-life smart devices, and second, that is publicly available. For the fingerprinting method, a ML approach will be applied and the performance of various classification algorithms will be evaluated.

The rest of the paper is structured as follows. In Section II the related works on IoT security and privacy are presented, while in Section III the basics on traffic fingerprinting are introduced. Then, in Section IV an experimental test-bed as well as real-life IoT traffic analysis are outlined. Experimental results are enclosed in two sections: in Section V, IoT device identification is presented and in Section VI the detection of a user's activities based on IoT traffic is demonstrated. Next, Section VII has possible techniques to mitigate the risks related to the presented threats. Finally, Section VIII concludes our research and outlines future work.

II. RELATED WORK

The IoT has not yet reached a common standard despite some efforts, for instance, from IEEE, and is considered highly heterogeneous and enabled by a variety of different technologies on each architecture level (wireless networks communication, cloud computing, etc). This, in fact, is one of the reasons for the security issues we are observing today. The other is that IoT devices have limited resources, like processing power or battery life, so making the state-of-the-art security solutions not suitable for them. In the reminder of this section we describe the main IoT security and privacy problems.

A. IoT SECURITY PROBLEMS

Many IoT devices have their network interfaces exposed to the public, so that it is easy to identify and reach them using open-source tools like the Shodan.io search engine.[3] Consequently they become an easy target for cybercriminals. Numerous examples of successful attacks have been demonstrated throughout the years that drew publicity to the topic of the IoT security. As an example, the Trendnet home security camera was found to enable remote attackers to access its live video stream using only its IP address.[4] Even with password protection applied, the credentials were sent over the network and stored in plaintext form.

Miller and Valasek showed how to remotely take control over a Jeep car through its on-board computer [4]. They were able, for instance, to change the volume of the radio, turn on the windshield wipers, and even control the brakes of the vehicle. For their hack, they used vulnerabilities such as exposed ports in the car systems and guessable access parameters. Another highly alarming security issue concerned medical devices. In 2016 Muddy Water's Research revealed in their report that St. Jude Medical's implantable cardiac devices (such as pacemakers) have security flaws allowing an attacker to access the devices and then deplete the battery or administer incorrect pacing or shocks.[5]

Even though IoT devices are sometimes considered the ''Internet's least powerful hosts'' [3], they can also be used to become members of a large botnet and then participate in DDoS (Distributed Denial of Service) attacks on other devices and services. One of the most recent and large-scale attacks of this kind was launched by the Mirai botnet, which reportedly infected up to 600k devices to successfully take down many online services, such as the krebsonsecurity.com website and French hosting service OVH [3]. The latter is said to have experienced a peak rate of at least 1.1 Tbps, which broke the record for DDoS attack volume. The strategy that Mirai used to infect the devices was very simple: it used a list of the most common default passwords for home routers, network-enabled cameras and digital video recorders, which among IoT devices had the fewest protection measures.

Hence, many researchers took the challenge to summarize and classify the security threats that the IoT is facing – a common approach for identifying potential risks and attacks at every layer of the IoT reference model is proposed in [10], [11].

B. IoT PRIVACY PROBLEMS

There has been a heated discussion on the privacy implications of using the devices. The main privacy concerns include [11]:

- *Identification* – the threat of associating an identifier (e.g., persistent) with a person and data related to them. This can be, for instance, their name, phone numbers, photos or their device's unique fingerprint. It can enable other privacy threats, such as profiling and linking.
- *Location-based tracking* – determining and recording a person's location through time and space (e.g., using GPS service or network address).
- *Profiling* – the threat of compiling information (from multiple sources) to infer the behavior and interests of a user. Profiling methods are used in online

[3]https://www.shodan.io

[4]https://www.wired.com/2012/02/home-cameras-exposed

[5]http://d.muddywatersresearch.com/content/uploads/2016/08/MW_STJ_08252016_2.pdf

recommendation systems and advertisements. However, they may sometimes lead to disclosing information considered as sensitive by a user (sexual orientation, medical conditions) [12] or to price discrimination (offering higher prices than usual based on some information possessed by a customer) [13].

- *Inventory attack* – collection of information about existence and characteristics of objects and devices (computers, phones, IoT) that belong to a user without their consent. This threat is closely related to fingerprinting techniques (i.e., device fingerprinting). Information on inventory can be used, for example, by burglars for targeted break-ins at private homes or offices or by law enforcement for unwarranted searches.
- *Linkage* – refers to combining data from different sources to reveal information previously unavailable using separate sources. This threat is transparent to users because they are usually not aware of the contexts and times in which the different data was collected and how they can be used together. Linkage also increases the risk of re-identification of anonymized data (e.g., health data [14]).

A great number of concerns about user's data safety have been reported. One example is a case of soldiers who accidentally revealed a secret military base's location via their fitness trackers.[6] This was caused by the application (Strava) cooperating with wearable devices that published a "heat map" of users' movements. In locations like Afghanistan, where fitness trackers were mostly used by foreign soldiers and not the locals, one could easily identify the base's location solely from this information.

In [15] the authors examined a personalized light switch (which adapts its brightness and color settings depending on a specific user) and remote watering systems. Among other vulnerabilities (e.g., remote device exploitation) they point out some privacy issues such as leakage of personally identifiable information and emanation of information relative to user location. They found out, for example, that in the case of the light switch system, unique identifiers of associated users are being broadcasted by the devices in plaintext, which allows their tracking, for instance, in an office environment. In addition, the watering system did not encrypt communication, which made eavesdropping possible.

Privacy problems are also no stranger to IoT toys. Some teddy bears or Barbie dolls allow voice recording and sending voice messages (e.g., from parents to children) or generating automated responses based on a child's input, using connectivity to a remote server where data is stored. According to mass-media,[7] [8] these toys can easily be hacked and turned into surveillance devices.

[6]https://www.nytimes.com/2018/01/29/world/middleeast/strava-heat-map.html

[7]https://www.forbrukerradet.no/siste-nytt/connected-toys-violate-consumer-laws

[8]https://www.vice.com/en_us/article/qkm48b/how-this-internet-of-things-teddy-bear-can-be-remotely-turned-into-a-spy-device

Even in an area as sensitive as medical devices, a user cannot be entirely sure that security measures were undertaken to protect their personal data. [16] has shown that many such devices transport collected data (pictures, user identifiers, actions performed) in plaintext, which makes it easy to intercept them by an eavesdropper. The common solution to this privacy problem is applying encryption to the data transfer, which the authors suggest. However, further research in this topic shows that even in this case it is possible to extract some information from the traffic: the number and the type of utilized devices, infer the user's actions and behavior patterns. In [17] the authors examined some popular smart home devices that may raise privacy concerns in terms of the traffic they generate. They found that they can be identified by a remote attacker using unencrypted Domain Name System (DNS) queries. Moreover, their traffic traces exposed characteristic transmission rates, which revealed the actions currently performed by a device or the user.

In [9], Bluetooth traffic coming from fitness trackers was analyzed. The authors collected the traces using a ComProbe Bluetooth Protocol Analyzer (BPA) device in a gym and concluded that a user can not only be tracked using a constant identifier of their device, but also in a group of people based on their wearable device measurements.

III. TRAFFIC FINGERPRINTING

A traffic analysis attack can be defined as network traffic monitoring to identify useful patterns in the traces that can be used for defense and detection of security risks (such as botnet traffic) [18], forensics (to identify a criminal or if a crime was committed) [19] or for malicious purposes like attacks on privacy. One of the traffic analysis methods is called traffic fingerprinting, which is often described as a passive privacy related side-channel attack [5]. It stands for comparing and matching features extracted from the network traces (such as packet sizes, distribution or different statistical measures) with the known models (fingerprint). Fingerprinting techniques often use a ML-based approach – classifiers, to achieve better accuracy when recognizing traffic type.

A. FINGERPRINTING DOMAINS

There are multiple traffic fingerprinting types that aim either to identify the user and the device they are using or the activity they are performing (e.g., applications used or websites accessed). This section summarizes the existing solutions in this field.

1) WEBSITE FINGERPRINTING (WF)

The WF attack aims at inferring which website a user has accessed by observing their network traces, even in the presence of anonymization and obfuscation techniques. The common scenario assumes that an Internet user hides their browsing activities (destination IP) behind some proxy service or employs a different anonymization technology and a passive observer located in the user's network (such as their ISP) tries to identify the websites being accessed. To achieve

this goal, they usually take advantage of the differences in website content loading that influence the fingerprint of a webpage. This technique is often used in forensics to estimate if a suspect accessed, for instance, a hidden service related to drug selling [19].

WF has been quite intensively researched in recent years. At first, researchers focused on fingerprinting encrypted traffic in the presence of a proxy or a VPN service. One of the first attacks of this type, such as [20], was based on an observation of the characteristic resource sizes per webpage (i.e., images and scripts). However, this solution was only effective for HTTP 1.0, that did not use persistent connections. Later work introduced website fingerprinting based on statistical features (such as inter-packet arrival time – IAT) and unique packet sizes [21]. Further research proved that the accuracy of the method can be increased when applying different ML classifiers, such as Naïve Bayes algorithm, firstly by [22]. The authors achieved around 90% accuracy working on OpenSSH protected traffic. The performance of the method was later improved (to about 97%) [6] by using a Multinomial classifier. However, upon the invention of TOR that introduced padding, packet sizes were no longer useful. As a result, most recent solutions focus on this service and employ more advanced fingerprinting methods.

The first to achieve a decent accuracy facing TOR were Panchenko et al. in [23] using a support vector machine (SVM) approach. Their method was able to recognize around 54% of URLs contained and around 74% in an open-world scenario (unlimited set of visited websites). These works drew a lot of attention in the research community and were followed by many others. Most current solutions, such as [6], reach an accuracy of ca. 90% in recognizing 100 monitored sites in an open-world scenario (around 5000 URLs not monitored). Their attack was still successful even against some fingerprinting countermeasures, however, as shown in [24], in this case the recognition level decreased significantly. The authors demonstrated that the accuracy of website fingerprinting can drop by 40% during the first nine days and goes down to zero within 90 days for Alexa Top 100 websites. The dynamism of the web environment is the real challenge for creators of new fingerprinting methods.

2) APPLICATION FINGERPRINTING

Another frequently discussed domain concerns applications, i.e., identifying traffic from various online services, including Voice over IP (VoIP), video and audio streaming and also mobile applications. Two approaches prevail in the literature: the first one focuses on classifying just the type of traffic (e.g., online games, download, VoIP, video streaming and Peer2Peer, etc.), the second aims at identifying the particular application generating it (e.g., Skype, Spotify, Youtube and BitTorrent, etc.) [25].

In the case of fingerprinting streaming applications, researchers usually take advantage of their bursty traffic profile. In [26] the authors use burst patterns to analyze different video streams (YouTube, Netflix, Vimeo and Amazon, etc.) using convolutional neural networks to identify which content exactly the user is currently watching.

A recent trend in the fingerprinting domain is focused on mobile devices and applications. Since a lot of mobile applications send their data using the HTTP(S) protocol, this particular domain shares similarities with website fingerprinting. However, the authors of [27] state that fingerprinting the traffic originating from a mobile device is a more complicated process because of the text-based APIs being used in smartphones that remove the rich features of HTTP normally present in traditional browsing traffic. However, they show the ability of a passive eavesdropper to identify smartphone apps by fingerprinting encrypted packet traces. They tested their method on 110 Android applications and, notably, achieved 96% accuracy even after six months after preparing their fingerprints. [28] checked to what extent the attacker is able to identify specific actions performed by a user via mobile apps (such as writing tweets using Twitter and adding comment on Facebook, etc.). For most activities they achieved around 95% accuracy.

3) DEVICE FINGERPRINTING

Fingerprinting devices aims to extract information from the network traffic, based on which it is possible to identify them. It is feasible due to tolerances in the manufacturing process and differences in hardware and software implementations. They can either capture class characteristics, which are emitted by all devices that share the same specification, or individual ones [19].

Device fingerprinting found its application in online tracking mechanisms and is used by many advertising companies to keep track of the user's activities, such as their browsing behavior, based on their phone/computer fingerprint, even cross-device [12]. To identify a single device a variety of features are extracted from the network traffic, such as: user-agent (browser type), fonts, languages, time zone, etc.

Device fingerprinting is also considered to be an authentication factor. A paper [29] focuses on device fingerprinting in terms of security enhancing and protection of a wireless network from unauthorized, malicious nodes. The authors rely mostly on the features of the physical traffic layer, such as clock skew from timestamps, frequency offsets, phase offsets and then apply supervised (white-list) and unsupervised learning classifiers to detect illegitimate nodes on the network. Another recent example of such an approach proposes a 2-stage classification model for the detection of an intruding device in the local network based on (among others) port numbers, domain names, cipher suites and flow volumes [7]. The authors test their model in a large environment of 28 devices to achieve over 99% accuracy.

Note, that fingerprinting techniques can be also applied to launch a privacy attack, which is discussed in the next subsection.

4) PRIVACY-ORIENTED IoT DEVICE FINGERPRINTING

Such an attack is presented in a series of publications [17], [30]. The authors show a scenario, in which a last-mile

passive observer can invade a user's privacy by identifying the IoT appliances they have at home using either DNS queries or traffic rates which, as they demonstrated using 2-feature vectors and k-NN classifier, achieved 95% accuracy. Moreover, it is also possible for some devices to expose the user's behavior patterns through observing peaks on traffic rate graphs. Some defense measures (such as traffic shaping) were also proposed and tested. The authors emphasize, however, that further research should be performed regarding the application of ML for privacy-based attacks in search of, e.g., different traffic features that would improve attack feasibility and accuracy even in larger environments (experiments were performed on six devices).

Another approach was presented in [31] where the authors, rather than on device recognition, focus more on the IoT activity detection by an attacker able to sniff on wireless traffic (such as ZigBee, WiFi or Bluetooth). They extracted and used a large number of statistical time series features to firstly – detect if an activity has been triggered (using binary classification) and secondly – recognize this activity (multiclass classification). This way they achieved on average above 90% accuracy (though with a low value of support metric – only a few ''positive'' samples). Still, the IoT traffic analysis attacks are not, to the best of our knowledge, frequently covered in the literature and, thus, this still constitutes an open research problem.

5) BEHAVIORAL FINGERPRINTING

The last discussed domain of fingerprinting focuses on the identification of a single user through their online behavior. This set of techniques allows the linking of multiple browsing sessions from the same user as well as to distinguish the sessions of different users based only on the characteristic usage patterns [19].

In [12] the authors show how a person can be profiled based on their web browsing history that can be shared with third parties through the multiple tracking mechanisms present on a website. It proves that by having access to this resource one can extract sensitive information about users, such as political opinions, gender, religious beliefs and health related data. In [32] it is shown that a user can be accurately identified based on their browsing habits. The authors performed their experiments on an anonymized dataset collected from 3862 students in a two month period, and for each of the users they extracted a fingerprint comprising a set of DNS queries they issued and their access frequency. Then a Multinomial Bayes classifier was applied on the other browsing sessions. The method proved to yield between 70% accuracy (for 24h browsing sessions of 3000 users) and 90% (for 100 concurrent users).

Later, in [19] the authors stated that this method can be useful for ascription of activities (i.e., criminal) in forensics investigations. Then, [32] showed that it was possible to identify a person through the analysis of traffic coming from their wearable fitness trackers. The authors noticed a correlation between Bluetooth traffic features (e.g., payload size, data rates) and measurement values (e.g., for acceleration) and used this observation to distinguish between single persons in the same location (i.e., gym, office). For classification they use Decision Trees, achieving around 89% accuracy for identification within a group of five people. Exposing the user's behavioral profile through networked devices remains one of the biggest privacy concerns with current technologies.

B. NETWORK TRAFFIC CLASSIFICATION

When encryption started to be commonly used in data transport it made the task of network traffic analysis and anomaly detection far more challenging, and thus there is an increasing need to invent more advanced analysis methods that are able to identify the type of application used via traffic fingerprinting. The authors of [33] mention two groups of classification approaches. The first one comprises the payload-based methods, while the second group focuses on the feature-based techniques, most of which use ML algorithms. This section provides a brief overview of these techniques and their specific types and applications.

1) PAYLOAD-BASED TRAFFIC ANALYSIS

This group of methods uses differences in packet formats to classify traffic streams. They inspect the payload of a packet in search of specific protocol related structures together with applying regular expressions or string matching algorithms to determine information about the traffic type.

This kind of classification method was used in the PACE system.[9] It comprised a commercial library, which was able to classify many obfuscated protocols and applications using behavioral and statistical analysis and augmented heuristics. Another example is Cisco Network Based Application Recognition (NBAR)[10] that is used mostly on Cisco routers for quality and security purposes. One open-source example of a payload-based classifier is nDPI [34] that was originally derived from PACE. Its huge drawback is that it has little functionality for analyzing encrypted traffic limited to the decoding of the hostname from the certificate in SSL. It also relies only on the eight first packets of each connection to classify the analyzed traffic.

However, the application of these techniques in the literature does not go beyond the identification of obfuscated protocols and analysis of unencrypted traffic, also due to performance and the more efficient modern payload obfuscation methods [35] and this has been used only in complex commercial systems.

2) FEATURE-BASED TRAFFIC ANALYSIS

This group of methods specializes in encrypted traffic analysis. Contrary to the payload-based approach, they rely on the characteristics of packet traces (on a flow or a packet level), such as length, direction and statistical values, etc.

[9] http://www.ipoque.com/en/products/pace

[10] https://www.cisco.com/c/en/us/products/ios-nx-os-software/network-based-application-recognition-nbar/index.html

for classification purposes. In the analysis of encrypted traffic every ML algorithm has a different approach to sorting and prioritizing sets of features, which leads to different dynamic behaviors during training and classification phases. Frequently used ML-based methods in this group are Naïve Bayes, k-NN, Decision Trees, SVMs and Neural Networks.

Based on the literature review it can be seen that the vast majority of the methods applied for traffic fingerprinting nowadays are based on ML algorithms as it was proved to be more effective when dealing with encrypted or obfuscated traffic. It is also worth mentioning that the performance (in terms of accuracy) of ML classification does not only depend on the chosen algorithm but also on the selected features, the dataset itself and on the anonymization method applied on the traffic.

C. NETWORK TRAFFIC FEATURES

Fingerprinting techniques take advantage of the network traffic characteristics, which enables identification of patterns in the packet traces. In most solutions, a combination of different traffic characteristics is used to optimize classification accuracy.

The features most often used in fingerprinting involve:

- *Packet size* – most applications use specific sizes of payloads, which enabled their profiling. One of the first such approaches was used in [21] for the website fingerprinting purpose. However, this feature is not helpful when faced with packet padding mechanisms (e.g., used by TOR).
- *Packet direction* – information on whether the traffic is incoming or outgoing.
- *Packet length frequency* – the number of occurrences that a specific packet size has in the network trace. It can be expressed as a mean or a standard deviation [36].
- *Packet ordering* – depends on network conditions: it may vary facing problems with bandwidth or latency, and it may be affected by pipelining and changing parameters (in HTTP connections), etc.
- *Inter-packet arrival time* (IAT) – may prove to be very useful for fingerprinting purposes to infer the logical relationship between packets in the trace [21], also by measuring the round-trip-time value (RTT) [36]. It is not sensitive to packet padding and does not require stream partitioning [37]. However, this feature can also be influenced by network performance issues.
- *Packet count* – can be expressed in many different ways, such as the count of total/incoming/outgoing packets and ratios between them, etc. [38].
- *Burst size* – total size of the burst's content [23], [38] or *burst count* – number of packets contained in a burst [38], [39], where the burst is defined as a sequence of packets sent in one direction that lie between two packets sent in the opposite direction.
- *Traffic data rates* – measurement of the speed of upstream and downstream data, which was successfully applied to investigate IoT devices [8] or in HTTP traffic fingerprinting [31].

IV. EXPERIMENTAL TEST-BED AND REAL-LIFE IoT TRAFFIC ANALYSIS

The recent popularity of the application of the ML concept for problem solving has led to a lot of datasets being publicly available with network traffic traces among them. However, as mentioned beforehand, the research on IoT fingerprinting is rather scarce with only a few works in this field, e.g., [24], [30], [31]. Moreover, it is common that authors do not share publicly the dataset they used to conduct research. Unfortunately, the efforts to contact them and obtain the data were unsuccessful.

Publication [7] is, to the best of the authors' knowledge, the first one to publish their dataset for research purposes. The authors made available traffic traces collected in a smart environment in the span of two weeks with around 28 IoT devices including smart plugs, motion sensors, IP cameras and health monitors etc. However, as this data proved to be useful for extending the first problem addressed in the present study (device identification), it does not provide any information on the device working modes in the traces. Therefore, labeling the date for the task of activity recognition is impossible unless one is equipped with exactly the same appliances.

To overcome the shortcomings of the available traces and solve the second problem researched in this paper (state recognition), there was a need for a self-generated dataset.

A. EXPERIMENTAL SETUP

A dedicated smart home laboratory environment was set up enabling the collection and analysis of the generated network traffic. The test-bed comprised five common, low-cost, easily available smart devices: *(i)* Xiaomi-Philips Smart Lightbulb (device No. 1 in Fig. 1);[11] *(ii)* Neo Motion Sensor (2);[12] *(iii)* Nexlux Wifibulb (3)];[13] *(iv)* Dlink Motion Sensor DCH-S15 (4)[14] and *(v)* Broadlink Smart Plug. (5)[15] A schematic illustration of the test-bed architecture is presented in Fig. 1.

All of the smart appliances communicated over a Wi-Fi connection with a Raspberry Pi computer serving as a smart home gateway/router and were controlled via a dedicated mobile application. The Raspberry 3 Pi model B+ device operating on the most recent Raspbian Stretch (a Debian-like Linux distribution) version was set up as a wireless access point using its built-in Wi-Fi antenna (802.11b was used for compatibility) and *hostapd* (host access point deamon). In addition, the *dnsmasq* service was configured to serve as a DNS and Dynamic Host Configuration Protocol (DHCP) agent. Finally, an *iptables* services rule was set up to configure the Network Address Translation (NAT) between the

[11] https://item.mi.com/1172100033.html

[12] https://www.szneo.com/en/products/index.php?id=58

[13] https://www.amazon.com/Nexlux-Cellphone-Equivalent-Compatible-Assistant/dp/B07BGWKD43

[14] https://eu.dlink.com/pl/pl/products/dch-s150-motion-sensor

[15] http://www.ibroadlink.com/sp3/

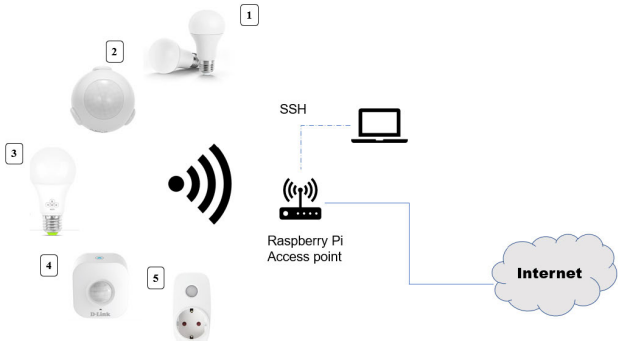

FIGURE 1. Simplified schematic drawing of test-bed architecture (devices are described in text).

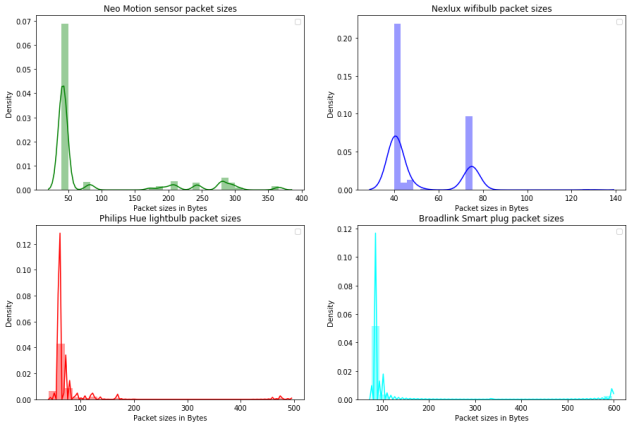

FIGURE 2. Packet size distributions in traffic samples of IoT devices. Bars plots are histograms of packet sizes (grouped in bins), while line plots represent normal distributions (values clustered around mean value of each bin).

Wi-Fi interface with AP and the Ethernet interface connected to the Wide Area Network (WAN) so that the connected devices had access to the Internet.

The Raspberry Pi acted as a traffic monitor for the IoT devices and was used to collect and save network traffic, by capturing it using the *tcpdump* tool. This data in the form of packet capture (PCAP) files was transferred via the Secure Shell (SSH) service running on the Raspberry Pi. Packet traces were collected over two weeks in both unconstrained device usage (simulating realistic scenario) and a series of controlled experiments. The data were analysed on a HP EliteBook G5 with CPU: Intel Core i7-8650U, 32GB RAM and GPU: Intel UHD Graphics 620. The Python toolbox *scikit-learn*[16] was used for experiments with ML.

B. REAL-LIFE IoT TRAFFIC OBSERVATION AND ANALYSIS

The observation of the collected packets for different devices showed their characteristic traffic profiles. The difference between them was noticeable especially in terms of packet size values (see distributions in Fig. 2), packet IAT values and transmission rates.

[16]https://scikit-learn.org

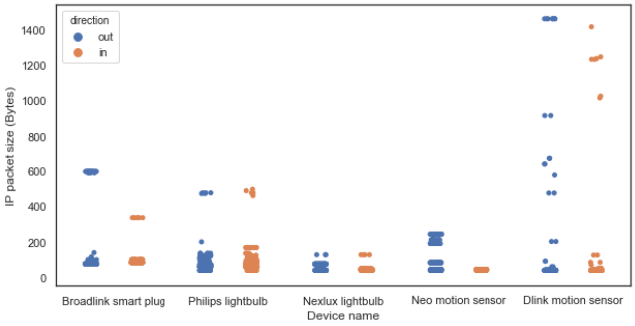

FIGURE 3. Packet size distribution per packet direction (outgoing in blue, incoming in orange).

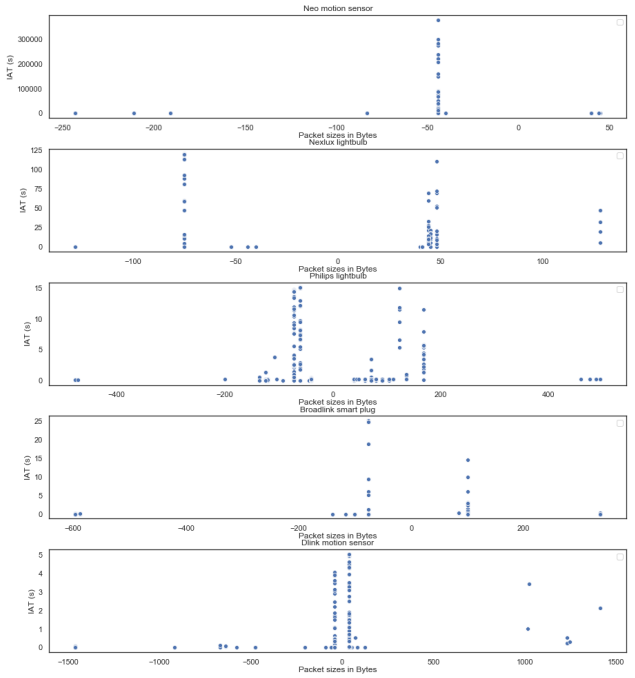

FIGURE 4. Scatterplots for all five devices showing dependence of packet size and IAT for ca. 400 packets. Negative size values are outgoing (upstream) packets while positive values are incoming (downstream) traffic.

As Fig. 3 shows, the packet size values used by each of the devices fall in specific ranges. Another observation was that the lengths were usually specific for incoming and outgoing traffic (Fig. 4). Among the tested devices, only in the lightbulbs was the packet size parameter distributed similarly for each transmission direction.

Most of the devices sent some heartbeat (a short periodical exchange) traffic when they were idle, while during their state change (e.g., motion detection, turning on/off and switching working modes) a sudden rise in the transmission rate was detected (Fig. 5). Moreover, such an event was usually closely linked to a distinctive sequence of exchanged packets.

Each tested appliance sent their data to the remote servers under multiple IP addresses. In most cases the IP of the destination domain was changed very dynamically. The DNS responses to requests raised by the devices were not encrypted

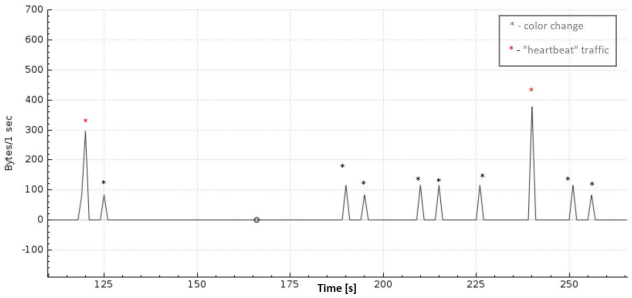

FIGURE 5. I/O graphs presenting overall traffic as per-second rate for Nexlux lightbulb.

so a passive eavesdropper could sniff and record domain names and their IP mappings. This could allow them to identify to which device the captured traffic belongs. However, if a vendor uses non-suggestive or random domain names or implements encrypted DNS in the solution, this information becomes useless for an adversary.

C. ADVERSARY MODEL

In our experiments we consider two types of attackers:

- *Local network intruder* – an adversary gains access to the victim's LAN, for example, through hacking their access point (or even gets this access in a legitimate way) and is able to sniff the traffic generated by connected devices.
- *Remote attacker* – an adversary is able to sniff the network traffic of the victim's household while being outside of their LAN. This could also be an Internet Service Provider or someone with access to the last-mile connection to the victim's home or even an adversary who comes into possession of a node on the routing path in the WAN through, for example, a BGP hijacking attack [40].

The difference between both adversaries is the visibility of the traffic details. For instance, an attacker operating in a LAN would be able to distinguish the captures for individual devices using their locally assigned IP address while for one outside the local network they would come across these addresses and ports rewritten to one value by a router's NAT. This study aims to create a universal fingerprinting model that can be used by both types of adversaries. Therefore, it will take advantage of the information that is visible on both sides of the network.

D. TRAFFIC CLASSIFICATION MODEL

To learn a device's state and user's activity as a consequence, the collected traffic had to undergo a 2-stage classification process. In the first step of the algorithm an adversary had to identify which device they were dealing with based on the network traces they gathered. We implemented a single multi-class classifier and ran experiments with three supervised classification methods: k-NN, Decision Trees and Random Forests.

The k-NN algorithm is a fairly easy classifier, which have been used since the 1950s. It makes its decision based on the

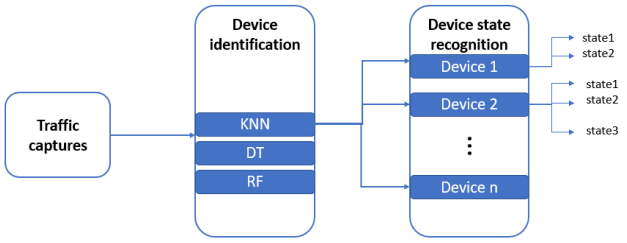

FIGURE 6. Overview of full fingerprinting attack model.

k closest samples from the training set. In our experiments we used $k = 5$. As for the Decision Trees, we used the Classification And Regression Trees (CART) classifier, which constructs binary trees by choosing features and threshold values which maximize information gain at each node. Random Forests are ensemble classifiers which use a cohort of simpler classifiers, in this case – Decision Trees. Random Forests tend to be less prone to overfitting than simple classifiers. In our experiments we used Random Forests with 100 trees. Number of trees in Random Forests, as well as neighbors in k-NNs were selected heuristically.

After the flow was recognized, it underwent another multi-class classification to determine the state of the device. During this phase, a separate classifier (another Decision Tree) was trained separately for each device. The overview of the whole attack is presented in Fig. 6.

E. DATA PREPROCESSING

To generate the ground truth for the training ML models, the traffic traces of individual devices were separated based on the local IP addresses. However, what a last-mile adversary would recognize was a distinct IP address of the remote server communicating with a particular IoT device. The collected traffic traces were therefore split by the remote IP addresses into flows containing both outgoing and incoming traffic from this IP. Then, using *tshark* (terminal-oriented version of Wireshark) the PCAP files were transformed into CSV files (see Fig. 7) containing vectors with timestamp, source and destination IP addresses, packet length and IAT values.

V. DEVICE IDENTIFICATION THROUGH TRAFFIC FINGERPRINTING

The first step in the fingerprinting attack aims at identification through statistical traffic analysis which devices the victim is using. This section describes features extraction and engineering done to achieve this goal and finally presents results obtained using ML algorithms for various setups.

A. FEATURE EXTRACTION FROM NETWORK TRAFFIC AND FEATURE ENGINEERING

While observing IoT traffic characteristics (Section IV-B) it became clear that one of the main features differentiating the devices are the packet lengths and the transmission rates. It was also noted that each smart appliance had a specific model of communication with the remote server (i.e., different number and sizes of requests and responses), which could

FIGURE 7. Schematic representation of preprocessing flow.

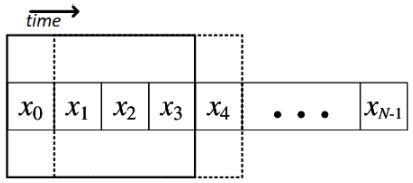

FIGURE 8. Sliding window mechanism (x_n – observations – network packets in this case).

TABLE 1. Summary of traffic from five laboratory devices contained in self-generated dataset.

Device	Number of samples
Broadlink smart plug	20405
Philips lightbulb	17412
Nexlux lightbulb	5651
Neo motion sensor	5566
Dlink motion sensor	15683

be captured if observing the device behavior within a time period. Therefore we proposed to transform single packet features (such as size, direction) into a flow-based feature space, so we calculated traffic statistics (e.g., mean, standard deviation) over a fixed-length sliding window (Fig. 8). Its length was chosen experimentally.

The features that were finally extracted in each time window included: *(i)* mean IP packet length (bytes), *(ii)* transmission rate (bytes/sec.), *(iii)* mean IAT value, *(iv)* standard deviation of the packet lengths, *(v)* outgoing/incoming traffic transmission rate (bytes/sec.) and *(vi)* mean size of outgoing/incoming packets (bytes).

Following the feature extraction process, the samples were labeled with corresponding device names and saved to CSV files. The data gathered over two weeks are summarized in Table 1.

The dataset is not balanced (it has an uneven number of samples per each class) because the devices generate different amounts of traffic, for instance, the Neo motion sensor communicates with the remote party only when a movement is detected. As Fig. 9 shows, the individual devices can be clustered quite effectively using information only about their packet sizes.

B. EXPERIMENTS WITH CLASSIFICATION OF IoT DEVICES

To recognize the IoT devices we used three ML algorithms: k-NN, Decision Trees and Random Forests. The data were split into the training and testing sets in the proportions 3:1. The k-fold cross validation (CV) technique with $k = 4$, implemented using the *StratifiedKfold* class from the *scikit-learn* library, was applied to optimally use the data. The CV was used for the algorithms' hyperparameters tuning

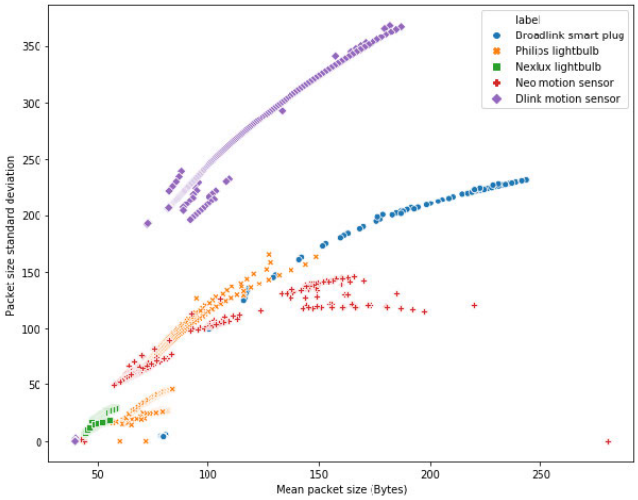

FIGURE 9. Scatter plot showing test IoT devices' packet sizes feature space.

TABLE 2. Classification performance given traffic aggregation in 200 sec. time windows using 4-fold CV (standard deviation across CV folds in brackets).

Model	Accuracy [%]	F1 score [%]	LogLos [%]
k-NN	98.83 (0.02)	98.16 (0.05)	-11.03 (2.17)
Decision Trees	99.80 (0.04)	99.65 (0.08)	-6.76 (0.92)
Random Forests	99.87 (0.02)	99.78 (0.05)	-1.57 (0.36)

as well (for example, the number of trees in the Random Forest algorithm).

C. CLASSIFICATION RESULTS FOR SMART HOME TEST-BED

Experiments performed in a self-created test-bed showed that it was possible to identify devices with almost 100% accuracy based on the statistical features of the network traffic (Table 2). Algorithms based on Decision Trees yielded the best performance, being ahead of the k-NN model by around 1%.

Next, to create a more realistic scenario, we added *non-IoT traffic* to the network. To simulate such a situation, the supplementary traffic was added coming from two laptops, a tablet and a connected printer, which was modeled as an additional class called non-IoT. In this scenario no major difference in accuracy was observed (Table 3). It confirms that the IoT traffic is very distinguishable from regular network activity. One major difference observed for all models was a decline in logarithmic loss. This means that introducing more diverse data (the non-IoT class) made the classifier less confident about its decisions.

TABLE 3. Classification performance for setup of five devices with non-IoT traffic given traffic aggregation in 200 sec. time windows, using 4-fold CV (standard deviation across CV folds in brackets).

Model	Accuracy [%]	F1 score [%]	LogLos [%]
k-NN	98.66 (0.06)	97.44 (0.12)	-12.8 (1.00)
Decision Trees	99.71 (0.03)	99.52 (0.03)	-8.67 (0.69)
Random Forests	99.81 (0.04)	99.7 (0.03)	-1.88 (0.40)

TABLE 4. Classification performance in IoT only environment given traffic aggregation in 300 sec. time windows using 4-fold CV (standard deviation across CV folds in brackets).

Model	Accuracy [%]	F1 score [%]	LogLos [%]
k-NN	92.46 (0.03)	84.64 (0.07)	-45.03 (0.75)
Decision Trees	92.09 (0.04)	84.51 (0.05)	-256.13 (1.43)
Random Forests	92.29 (0.04)	85.66 (0.08)	-63.16 (0.55)

D. IDENTIFYING IoT DEVICES IN LARGER SMART ENVIRONMENT

As previous sections showed, recognizing IoT devices in a small setup, even against regular user traffic was not a difficult task. However, due to the high diversity of the IoT solutions and devices, this model of fingerprinting needed to be tested and verified in a larger environment. To simulate such an environment, additional network traffic data was obtained from the dataset shared by the authors of [7]. Since the dataset contained aggregated captures per day, traces of individual devices were extracted using their MAC addresses. Only communication with the remote third parties was taken into account while all the LAN traffic was discarded. The same was done with the devices that had very few samples among the packet captures (e.g., the Blip blood pressure sensor). In total, additional traffic from 17 devices was extracted from the dataset, with the most data coming from Dropcam (390,202 samples) and the least data originating from the NEST smoke sensor (892 samples).

1) CLASSIFICATION RESULTS IN THE SMART HOME ONLY ENVIRONMENT

Training and testing the classifiers on a larger dataset resulted in an expected drop in overall accuracy. Yet it still remained at a high level of about 92%. The k-NN algorithm appeared to be the most accurate and the most confident in this scenario (Table 4). A window of 300 seconds in this case turned out to be optimal.

In comparison to the tests performed on the small set of devices described in Section V-C, we observed a significant decline in the logarithmic loss, especially for the Decision Trees classifier, where the score exceeded -2. By looking up the interpretation of this score on the LogLoss plot it can be noted that the classifier gave less than 10% confidence to the winning class, which is a poor score. The other classifiers introduced much better scores in this aspect.

On the other hand, by looking at the time statistics, it turned out that k-NN models required ca. 5 to 13 times more than the other two classifiers. It indicates its limited usability when dealing with large scale problems, when the model has to be updated frequently. However, we observed that the runtime of the algorithms did not depend on the aggregation window applied.

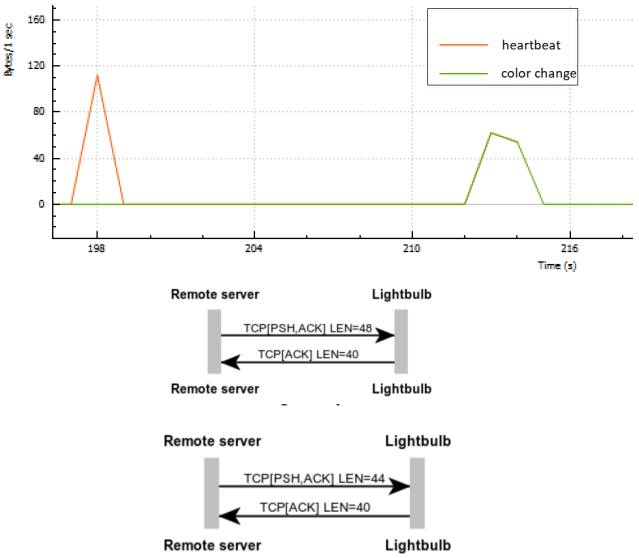

FIGURE 10. I/O and communication flow graph comparison of two different states (heartbeat traffic and change of color) of Nexlux lightbulb.

TABLE 5. Classification performance given traffic aggregation in 300 sec. time windows using 4-fold CV. Tests were performed in setup of 23 devices and non-IoT traffic (standard deviation across CV folds in brackets).

Model	Accuracy [%]	F1 score [%]	LogLos [%]
k-NN	92.27 (0.05)	85.15 (0.05)	-45.16 (1.1)
Decision Trees	91.88 (0.02)	84.51 (0.05)	-267.14 (0.6)
Random Forests	91.98 (0.04)	84.44 (0.07)	-65.23 (0.7)

2) CLASSIFICATION RESULTS IN PRESENCE OF NON-IoT TRAFFIC

Having added the non-IoT traffic we observed that it did not have a major impact on the classification accuracy, which remained on average ca. 92% (Table 5). The classification results for individual devices are shown in the form of a confusion matrix in Fig. 11. This demonstrates that although the vast majority of devices was recognized correctly, the model trained to also identify non-IoT traffic failed at classifying certain devices, including the Belkin motion sensor and switch and Withings smart scales – their traffic was usually confused with another device from the same manufacturer. The most likely reason for this behavior is that the producers designed very similar communication models for their smart appliances. The matrix also shows, that despite adding non-IoT traffic, none of the IoT devices were confused with the non-IoT class.

To summarize this section, we may conclude that the results obtained in various test cases showed unambiguously that by building an appropriate database of traffic fingerprints, an adversary can accurately predict which devices one possesses. Recognition problems appear when dealing with multiple devices from the same manufacturer that use similar

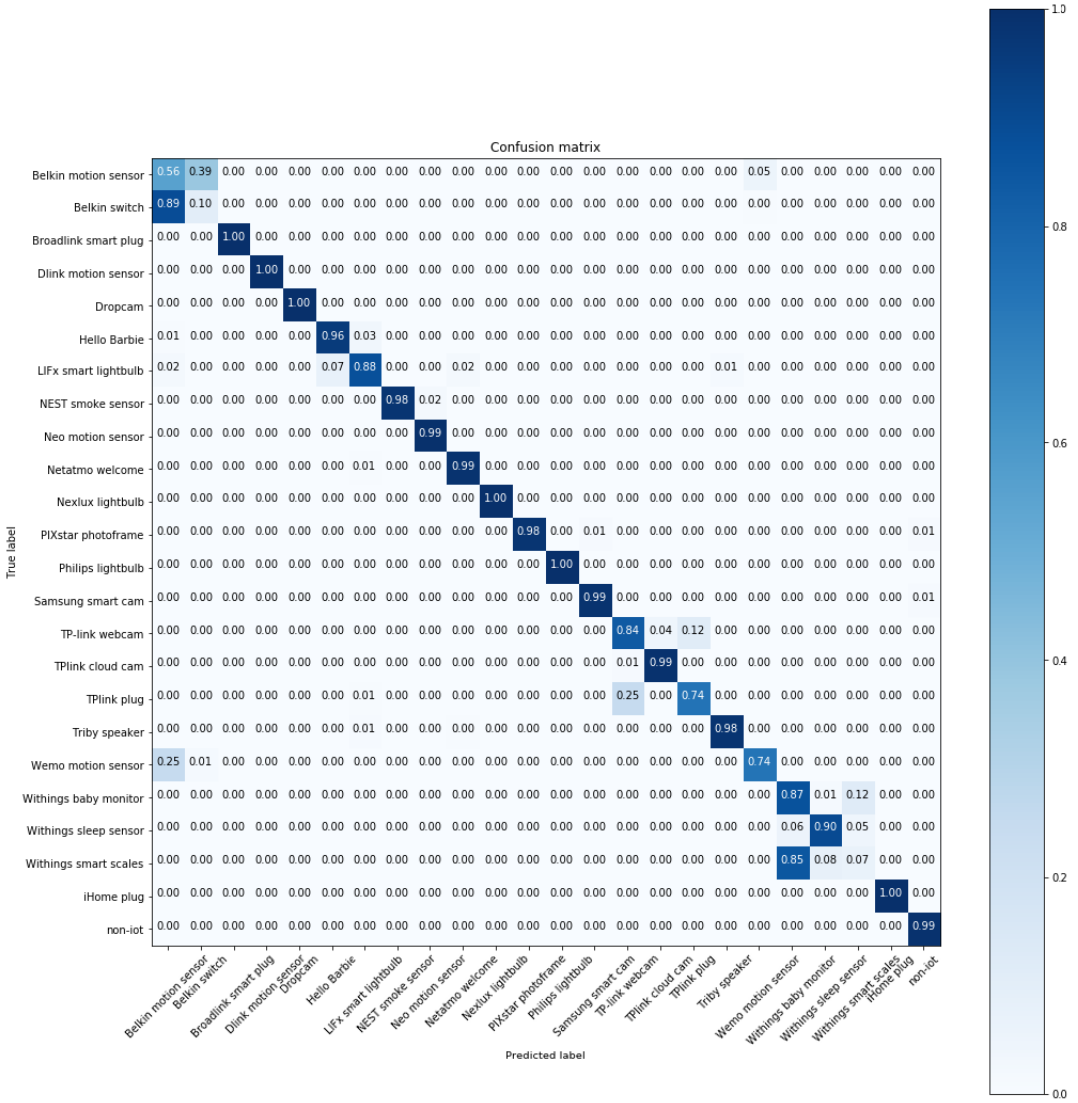

FIGURE 11. Confusion matrix presenting detailed results of Random Forests classification in the setup of 23 IoT devices and non-IoT traffic.

communication models. The experiments also showed high distinctiveness of smart device traffic against each other as well as against regular non-IoT network activities.

VI. USER ACTIVITIES DETECTION BASED ON IoT NETWORK TRAFFIC

Once an adversary identifies the devices used in a household by analyzing individual traffic traces, they may be able to find interesting patterns reflecting the states of those devices. This constitutes another privacy threat as the state changes are usually closely linked to the user activities and behavioral patterns. For instance, the detection of a motion or turning on/off of the light helps to determine whether a person is in or not. In this section we will show if using traffic analysis based on simple pattern recognition based on a ML model (Decision Trees) it is possible to recognize the state of devices in the test-bed.

A. FEATURE EXTRACTION AND ENGINEERING

The observation of network traffic generated by IoT devices under laboratory conditions led to a conclusion that each state change of an appliance was marked with a characteristic sequence of packets of distinct sizes. Therefore recognizing these sequences may lead to recognizing an activity related to the specific IoT device. For instance, in the case of the Nexlux lightbulb, the size of the request received by the sensor from the remote server differs significantly from the heartbeat traffic and action "color change" (Fig. 10).

We used the traffic data from the 5-device test-bed, as in the previous section. The traces shared by the authors of [7] could not be used for activity detection, as they provided no details on the device states or activities. What is more, to the best of the authors' knowledge, there are no publicly available IoT dataset with such information.

First, the raw packet traces (in the CSV form) were grouped into time windows. The size of these intervals needed to be adequate for an approximate duration of a state change. Arbitrarily a 3-sec. time window was applied to all the devices, apart from the Philips lightbulb, where a longer, 5-sec. window was required, as the activities for this device were visible in the network traffic significantly longer than for the other four devices. The feature vector was formed by transposing the sequence of packet sizes that fit into each time window. This way the classifier was able to make its decision based both on the packet sizes and their order. The labels of the vectors (e.g., on, off, motion detection, etc.) were assigned for each device manually either by comparing event occurrences in the corresponding mobile applications used to control and monitor a device or by recording times of events by hand.

For each of the devices the following states were distinguished:

- Nexlux lightbulb: idle, color change, on/off.
- Dlink motion sensor: idle, motion detection.
- Philips lightbulb: idle, on/off, user interaction with control application.
- Neo motion sensor: motion detection.
- Broadlink smart plug: idle, night light mode, status check via application, on/off.

The dataset was unbalanced because the samples labeled as idle were far more numerous than real activities. To slightly reduce this imbalance, a random part of the idle samples was ignored.

1) DEVICE STATE DETECTION WITH ML

The datasets created and labeled as explained in the previous sections were used to train two tree-based ML models: Decision Trees and Random Forests. The reason behind the choice of these algorithms is that they make a series of sequential decisions and would take into account the ordering of the values in vectors, which is important in the case of the studied scenario. 4-fold CV was used to avoid overfitting and optimally use the data.

The overall results achieved by both classifiers presented in Table 6 show that for all the tested devices it was highly effective to identify their states using the introduced method. In all cases the accuracy of the state change prediction exceeded 97% and the F1 score yielded at least 90%. During the observation of the network traffic from the IoT devices it was noticed that the Neo motion sensor generated packets only when it detected a motion (one state possible only) so the recognition would be perfect.

We found no significant difference between the performances of the Decision Trees and Random Forests, although in some cases the first classifier appears to achieve slightly better results. It proves that the traffic patterns related to different states can be recognized well using only one tree.

It is noticeable that ML algorithms were almost always correct at identifying the device's idle state. As expected,

TABLE 6. Overall evaluation results of device state recognition using 4-fold CV (standard deviation across CV folds in brackets).

Device	Decision Trees		Random Forests	
	Accuracy [%]	F1 score [%]	Accuracy [%]	F1 score [%]
Nexlux lightbulb	98 (2.4)	93.7 (6.6)	97.9 (2.3)	93.3 (6.9)
Dlink motion sensor	99.8 (0.08)	97.4 (1.0)	99.8 (0.08)	97.4 (1.0)
Philips lightbulb	98.7 (1.2)	91.7 (7.8)	98.6 (0.9)	89.7 (6.6)
Broadlink Smart plug	97.7 (0.9)	90.7 (0.5)	97.7 (0.5)	89.7 (2.8)

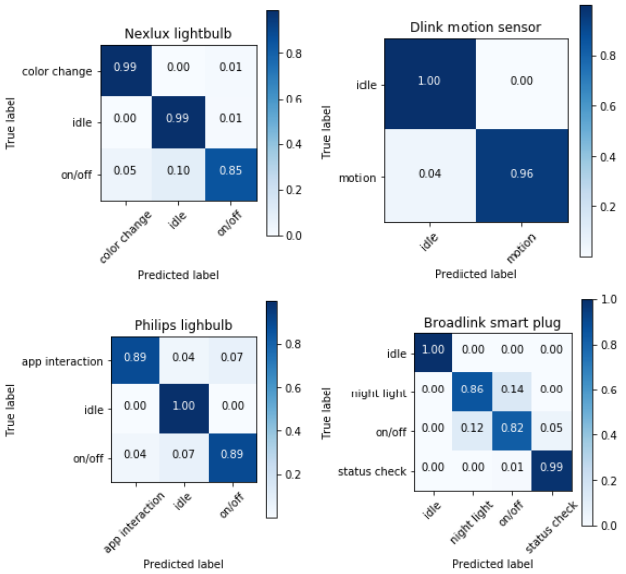

FIGURE 12. Normalized confusion matrices for device state recognition with Decision Trees model.

the recognition of the actual activities was more troublesome due to the diversity of the traffic patterns corresponding to them. This is clearly visible in the respective confusion matrices (Fig. 12). For instance, the smart plug activities (status check, on/off and night light) tended to be confused due to utilization of similar packet sizes.

To conclude, this section explored the possibility for an adversary to infer the states of IoT devices through traffic analysis and introduced a method for recognizing traffic patterns exhibited by the devices. The described experiments showed that each device state was identified with more than 80% accuracy, which indicates a significant risk for a user that their home activities could be recognized by a passive attacker.

VII. RECOMMENDATIONS ON RISK MITIGATION

The experimental evaluation presented in this paper clearly showed that both device and activity recognition can pose a privacy threat to the users of IoT devices. In this section we will suggest several techniques to mitigate this risk.

A. TRAFFIC TUNNELING

A user can decrease the risk of exposure of their sensitive information by routing all the smart home traffic through

TABLE 7. Detailed classification results for activity detection (results per device state) with decision trees model using 4-fold CV.

Device	State	Precision	Recall	F1 score	Sup.
Nexlux lightbulb	Color change	0.97	0.99	0.98	76
	Idle	0.99	0.99	0.99	618
	On/off	0.81	0.85	0.83	40
Dlink motion sensor	Idle	1	1	1	4002
	Motion detect.	0.94	0.96	0.95	89
Phillips lightbulb	App. inter.	0.92	0.89	0.91	27
	Idle	1	1	1	802
	On/off	0.86	0.89	0.88	28
Broadlink smart plug	Night Light	0.75	0.86	0.8	28
	Idle	1	1	1	396
	On/off	0.85	0.82	0.84	57
	Status check	0.99	0.99	0.99	358

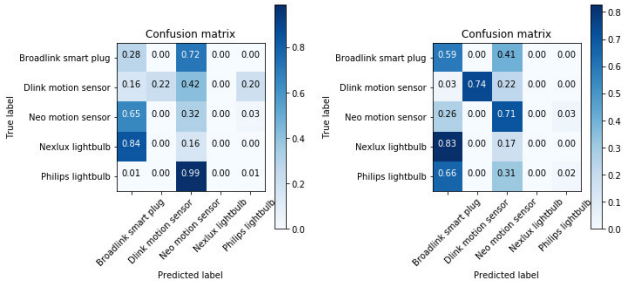

FIGURE 13. Confusion matrices presenting classification results on padded traffic datasets using Random Forests. Left matrix represents situation where only outgoing packets (traffic from devices) are padded, right matrix is where both incoming and outgoing packets are padded to constant size.

a proxy or a VPN server which would obscure the destination IP address of the network traffic. This way the attack risk is significantly lower as an attacker using simple methods would not be able to separate the traffic flows corresponding to individual IoT devices, which serve as the classification input. Still, the attack might prove successful if the IoT device is the only one at home, which is not a very realistic assumption.

B. TRAFFIC SHAPING AND DUMMY TRAFFIC INJECTION

Traffic shaping can be implemented by device manufacturers or by a third-party on a gateway router. There are already various methods of interfering with the communication models that allow hiding its context. One of the most straightforward examples is packet padding which relies on appending some irrelevant data at the end of the packets to obscure the flow characteristics. We applied this method to our data, by padding the packets up to a typical MTU size (1500 bytes) for the outgoing and incoming/outgoing traffic. The respective classification results are presented in Fig. 13.

In the first case the results showed a decrease in overall classification accuracy down to 34%. Only the traffic related to the two motion sensors was in general classified correctly (ca. 70%). When traffic in both directions was padded the total accuracy on the test set fell drastically to 14%. The main drawback of this method is, however, that it creates a traffic overhead and, potentially, introduces bandwidth and delay issues.

There exist, of course, more sophisticated methods of packet padding, including appending randomized amount of bytes to each or some of the packets. There are also solutions that tamper with the packet IAT values such as the one proposed in [36] or dummy traffic injection mechanisms such as in [31]. The application of more complex techniques of this kind will be considered in our future work.

VIII. CONCLUSION

In this paper traffic fingerprinting attacks concerning IoT devices have been shown and analyzed. First, the possibility of identifying individual smart home devices based on the statistical descriptors of the traffic was proposed. Various IoT setups were considered: a self-created test-bed with five off-the-shelf devices and a large environment simulated using publicly available IoT traffic traces. The results of the research proved that the IoT traffic is highly distinctive and allows a relatively easy identification of the IoT devices using statistical traffic features without the need for deep packet inspection. Interestingly, high recognition accuracy (above 90%) is also possible in the presence of the background traffic. We also proved that an adversary using ML methods can effectively learn the activities performed by a user by tracking state changes of the devices.

Contrary to most other studies, where ML techniques were used for cyber defense, we showed that they can also be a powerful tool in the hands of a cyber attacker. Information about possessed devices and their states may be used differently by adversaries, including reconnaissance of a future attack target – by a cyber or a physical attacker (a burglar may take advantage of knowledge about the users behavior). Moreover, a third party can make use of this information in profiling for unwanted targeted marketing. We also proposed how the risk posed by traffic analysis attacks can be mitigated.

This study assumed that no anonymization technique was used (such as proxy or VPN) that would unify remote party IP addresses and therefore disallow separation of the traffic flows, as in the attack described in this study. Studying the possibility of performing successful fingerprinting attacks in the presence of various defense methods will be considered in our future work. Finally, further research is needed to evaluate the described traffic fingerprinting method for more sophisticated IoT devices with complex network traffic patterns.

REFERENCES

[1] K. Ashton, "That 'Internet of Things' thing," *RFID J.*, vol. 22, pp. 97–114, Jun. 2009.

[2] A. Mosenia and N. K. Jha, "A comprehensive study of security of Internet-of-Things," *IEEE Trans. Emerg. Topics Comput.*, vol. 5, no. 4, pp. 586–602, Oct. 2017.

[3] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the Mirai Botnet," in *Proc. 26th USENIX Secur. Symp. USENIX Secur.*, Vancouver, BC, Canada, Aug. 2017, pp. 1093–1110. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis

[4] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, Aug. 2015. [Online]. Available: http://illmatics.com/RemoteCarHacking.pdf
[5] R. Spreitzer, V. Moonsamy, T. Korak, and S. Mangard, "Systematic classification of side-channel attacks: A case study for mobile devices," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 465–488, 1st Quart., 2018.
[6] T. Wang, X. Cai, R. Nithyanand, R. Johnson, and I. Goldberg, "Effective attacks and provable defenses for website fingerprinting," in *Proc. 23rd USENIX Secur. Symp. USENIX Secur.*, San Diego, CA, USA, Aug. 2014, pp. 143–157. [Online]. Available: https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/wang_tao
[7] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, "Classifying IoT devices in smart environments using network traffic characteristics," *IEEE Trans. Mobile Comput.*, vol. 18, no. 8, pp. 1745–1759, Aug. 2019.
[8] N. Apthorpe, D. Reisman, and N. Feamster, "Closing the blinds: Four strategies for protecting smart home privacy from network observers," 2017, *arXiv:1705.06809*. [Online]. Available: https://arxiv.org/abs/1705.06809
[9] A. K. Das, P. H. Pathak, C.-N. Chuah, and P. Mohapatra, "Uncovering privacy leakage in BLE network traffic of wearable fitness trackers," in *Proc. 17th Int. Workshop Mobile Comput. Syst. Appl.*, St. Augustine, FL, USA, 2016, pp. 99–104, doi: 10.1145/2873587.2873594.
[10] I. Andrea, C. Chrysostomou, and G. Hadjichristofi, "Internet of Things: Security vulnerabilities and challenges," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2015, pp. 180–187.
[11] J. H. Ziegeldorf, O. G. Morchon, and K. Wehrle, "Privacy in the Internet of Things: Threats and challenges," *Secur. Commun. Netw.*, vol. 7, no. 12, pp. 2728–2742, Dec. 2014, doi: 10.1002/sec.795.
[12] S. Zimmeck, J. S. Li, H. Kim, S. M. Bellovin, and T. Jebara, "A privacy analysis of cross-device tracking," in *Proc. 26th USENIX Conf. Secur. Symp.*, Berkeley, CA, USA, 2017, pp. 1391–1408. [Online]. Available: http://dl.acm.org/citation.cfm?id=3241189.3241297
[13] A. Odlyzko, "Privacy, economics, and price discrimination on the Internet," in *Proc. 5th Int. Conf. Electron. Commerce*, New York, NY, USA, 2003, pp. 355–366. [Online]. Available: http://doi.acm.org/10.1145/948005.948051
[14] K. El Emam, E. Jonker, L. Arbuckle, and B. Malin, "A systematic review of re-identification attacks on health data," *PLoS ONE*, vol. 6, no. 12, Dec. 2011, Art. no. e28071, doi: 10.1371/journal.pone.0028071.
[15] C. Kolias, A. Stavrou, J. Voas, I. Bojanova, and D. R. Kuhn, "Learning Internet-of-Things security 'hands-On,'" *IEEE Secur. Privacy*, vol. 14, no. 1, pp. 37–46, Jan./Feb. 2016, doi: 10.1109/MSP.2016.4.
[16] D. Wood, N. Apthorpe, and N. Feamster, "Cleartext data transmissions in consumer IoT medical devices," in *Proc. Workshop Internet Things Secur. Privacy*, 2017, pp. 7–12, doi: 10.1145/3139937.3139939.
[17] N. Apthorpe, D. Reisman, and N. Feamster, "A smart home is no castle: Privacy vulnerabilities of encrypted IoT traffic," 2017, *arXiv:1705.06805*. [Online]. Available: https://arxiv.org/abs/1705.06805
[18] C. J. Dietrich, C. Rossow, and N. Pohlmann, "CoCoSpot: Clustering and recognizing Botnet command and control channels using traffic analysis," *Comput. Netw.*, vol. 57, no. 2, pp. 475–486, Feb. 2013.
[19] D. Herrmann, K.-P. Fuchs, and H. Federrath, "Fingerprinting techniques for target-oriented investigations in network forensics," in *Sicherheit Sicherheit, Schutz und Zuverlässigkeit*, S. Katzenbeisser, V. Lotz, and E. Weippl, Eds. Bonn, Germany: Gesellschaft für Informatik eV, 2014, pp. 375–390.
[20] A. Hintz, "Fingerprinting websites using traffic analysis," in *Proc. 2nd Int. Conf. Privacy Enhancing Technol.* Berlin, Heidelberg: Springer-Verlag, 2003, pp. 171–178. [Online]. Available: http://dl.acm.org/citation.cfm?id=1765299.1765312
[21] G. D. Bissias, M. Liberatore, D. Jensen, and B. N. Levine, "Privacy vulnerabilities in encrypted HTTP streams," in *Proc. 5th Int. Workshop Privacy Enhancing Technol.* Berlin, Heidelberg: Springer-Verlag, 2006, pp. 1–11, doi: 10.1007/11767831_1.
[22] M. Liberatore and B. N. Levine, "Inferring the source of encrypted HTTP connections," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, New York, NY, USA, 2006, pp. 255–263, doi: 10.1145/1180405.1180437.
[23] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel, "Website fingerprinting in onion routing based anonymization networks," in *Proc. 10th Annu. ACM Workshop Privacy Electron. Soc.*, New York, NY, USA, 2011, pp. 103–114, doi: 10.1145/2046556.2046570.
[24] M. Juarez, S. Afroz, G. Acar, C. Diaz, and R. Greenstadt, "A critical evaluation of website fingerprinting attacks," in *Proc. 2014 ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, 2014, pp. 263–274, doi: 10.1145/2660267.2660368.
[25] J. Khalife, A. Hajjar, and J. Diaz-Verdejo, "A multilevel taxonomy and requirements for an optimal traffic-classification model," *Int. J. Netw. Manage.*, vol. 24, no. 2, pp. 101–120, Mar. 2014, doi: 10.1002/nem.1855.
[26] R. Schuster, V. Shmatikov, and E. Tromer, "Beauty and the burst: Remote identification of encrypted video streams," in *Proc. 26th USENIX Secur. Symp.*, Vancouver, BC, Canada, Aug. 2017, pp. 1357–1374. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/schuster
[27] V. F. Taylor, R. Spolaor, M. Conti, and I. Martinovic, "Robust smartphone app identification via encrypted network traffic analysis," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 1, pp. 63–78, Jan. 2018.
[28] M. Conti, L. V. Mancini, R. Spolaor, and N. V. Verde, "Analyzing Android encrypted network traffic to identify user actions," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 1, pp. 114–125, Jan. 2016.
[29] Q. Xu, R. Zheng, W. Saad, and Z. Han, "Device fingerprinting in wireless networks: Challenges and opportunities," *IEEE Commun. Surveys Tutr.*, vol. 18, no. 1, pp. 94–104, 1st Quart., 2016.
[30] N. Apthorpe, D. Reisman, S. Sundaresan, A. Narayanan, and N. Feamster, "Spying on the smart home: Privacy attacks and defenses on encrypted IoT traffic," 2017, *arXiv:1708.05044*. [Online]. Available: https://arxiv.org/abs/1708.05044
[31] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton, "Peek-a-boo, I still see you: Why efficient traffic analysis countermeasures fail," in *Proc. IEEE Symp. Secur. Privacy*, May 2012, pp. 332–346.
[32] C. Banse, D. Herrmann, and H. Federrath, "Tracking users on the Internet with behavioral patterns: Evaluation of its practical feasibility," in *Information Security and Privacy Research*, D. Gritzalis, S. Furnell, and M. Theoharidou, Eds. Berlin, Germany: Springer, 2012, pp. 235–248.
[33] P. Velan, M. Čermák, P. Čeleda, and M. Drašar, "A survey of methods for encrypted traffic classification and analysis," *Int. J. Netw. Manage.*, vol. 25, no. 5, pp. 355–374, Sep. 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nem.1901
[34] L. Deri, M. Martinelli, T. Bujlow, and A. Cardigliano, "NDPI: Open-source high-speed deep packet inspection," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Aug. 2014, pp. 617–622.
[35] S. Valenti, D. Rossi, A. Dainotti, A. Pescapè, A. Finamore, and M. Mellia, *Reviewing Traffic Classification*. Berlin, Germany: Springer, 2013, pp. 123–147, doi: 10.1007/978-3-642-36784-7_6.
[36] X. Cai, R. Nithyanand, T. Wang, R. Johnson, and I. Goldberg, "A systematic approach to developing and evaluating website fingerprinting defenses," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, 2014, pp. 227–238, doi: 10.1145/2660267.2660362.
[37] S. Feghhi and D. J. Leith, "A Web traffic analysis attack using only timing information," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 8, pp. 1747–1759, Aug. 2016.
[38] S. Li, H. Guo, and N. Hopper, "Measuring information leakage in website fingerprinting attacks and defenses," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, 2018, pp. 1977–1992, doi: 10.1145/3243734.3243832.
[39] K. Al-Naami, S. Chandra, A. Mustafa, L. Khan, Z. Lin, K. Hamlen, and B. Thuraisingham, "Adaptive encrypted traffic fingerprinting with bi-directional dependence," in *Proc. 32nd Annu. Conf. Comput. Secur. Appl.*, New York, NY, USA, 2016, pp. 177–188, doi: 10.1145/2991079.2991123.
[40] A. Kapela and A. Pilosov, *Stealing the Internet-a Routed, Wide-Area, Man-in-the-Middle Attack*. Las Vegas, NV, USA: Defcon, 2008.

MONIKA SKOWRON received the M.Sc. degree in telecommunications from the Warsaw University of Technology, in 2019. She has professional experience in cybersecurity and monitoring systems integration for the telecom sector. She currently works as a Cybersecurity Consultant. Her main interests include proactive cyber defense, digital forensics, and applications of data science in cybersecurity.

ARTUR JANICKI (Member, IEEE) received the M.Sc. and Ph.D. degrees (Hons.), in 1997 and 2004, respectively, and the D.Sc. degree (Habilitation) in telecommunications, in 2017, from the Faculty of Electronics and Information Technology, Warsaw University of Technology (WUT). He is currently a University Professor with the Cybersecurity Division, Institute of Telecommunications, WUT. His research and teaching activities focus on signal processing and machine learning, mostly in cybersecurity context. He is author or coauthor of over 60 conference and journal articles and a supervisor of over 50 B.Sc. and M.Sc. theses. He is a member of technical program committees of various international conferences and a reviewer for international journals in computer science and telecommunications.

WOJCIECH MAZURCZYK (Senior Member, IEEE) received the B.Sc., M.Sc., Ph.D. (Hons.), and D.Sc. (Habilitation) degrees in telecommunications from the Warsaw University of Technology (WUT), Warsaw, Poland, in 2003, 2004, 2009, and 2014, respectively. He is currently a Professor with the Institute of Computer Science, WUT. He also works as a Researcher at the Parallelism and VLSI Group, Faculty of Mathematics and Computer Science, FernUniversitaet, Germany. His research interests include bioinspired cybersecurity and networking, information hiding, and network security. He is involved in the technical program committee of many international conferences and also serves as a reviewer for major international magazines and journals. Since 2016, he has been the Editor-in-Chief of an open access *Journal of Cyber Security and Mobility*, and since 2018, he has been serving as an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY and as a *Mobile Communications and Networks Series* Editor for the *IEEE Communications Magazine*.