ScaNeF-IoT: Scalable Network Fingerprinting for IoT Devices

Tadani Nasser Alyahya
t.n.alyahya@soton.ac.uk
University of Southampton
Southampton, Hampshire, United Kingdom

Leonardo Aniello
l.aniello@soton.ac.uk
University of Southampton
Southampton, Hampshire, United Kingdom

Vladimiro Sassone
vsassone@soton.ac.uk
University of Southampton
Southampton, Hampshire, United Kingdom

ABSTRACT

Recognising IoT devices through network fingerprinting contributes to enhancing the security of IoT networks and supporting forensic activities. Machine learning techniques have been extensively utilised in the literature to optimise IoT fingerprinting accuracy. Given the rapid proliferation of new IoT devices, a current challenge in this field is around how to make IoT fingerprinting scalable, which involves efficiently updating the used machine learning model to enable the recognition of new IoT devices. Some approaches have been proposed to achieve scalability, but they all suffer from limitations like large memory requirements to store training data and accuracy decrease for older devices.

In this paper, we propose ScaNeF-IoT, a novel scalable network fingerprinting approach for IoT devices based on online stream learning and features extracted from fixed-size session payloads. Employing online stream learning allows to update the model without retaining training data. This, alongside relying on fixed-size session payloads, enables scalability without deteriorating recognition accuracy. We implement ScaNeF-IoT by analysing TCP/UDP payloads and utilising the Aggregated Mandrian Forest as the online stream learning algorithm. We provide a preliminary evaluation of ScaNeF-IoT accuracy and how it is affected as the model is updated iteratively to recognise new IoT devices. Furthermore, we compare ScaNeF-IoT accuracy with other IoT fingerprinting approaches, demonstrating that it is comparable to the state of the art and does not worsen as the classifier model is updated, despite not requiring to retain any training data for older IoT devices.

CCS CONCEPTS

• **Applied computing** → **Network forensics**; • **Security and privacy** → **Mobile and wireless security**; • **Computer systems organization** → **Sensor networks**; • **Computing methodologies** → **Online learning settings**.

KEYWORDS

Internet of Things (IoT), IoT device fingerprinting, device identification, passive scanning, scalability

ACM Reference Format:
Tadani Nasser Alyahya, Leonardo Aniello, and Vladimiro Sassone. 2024. ScaNeF-IoT: Scalable Network Fingerprinting for IoT Devices. In *The 19th International Conference on Availability, Reliability and Security (ARES 2024), July 30–August 02, 2024, Vienna, Austria.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3664476.3670892

ARES 2024, July 30–August 02, 2024, Vienna, Austria

ACM ISBN 979-8-4007-1718-5/24/07
https://doi.org/10.1145/3664476.3670892

1 INTRODUCTION

Security and forensics in IoT networks have become prominent nowadays, as IoT applications are pervasive in many environments, such as smart homes, university campuses and enterprise networks. Among the existing IoT security and forensics techniques, *network fingerprinting* allows to recognise individual IoT devices by analysing their traffic. More precisely, network fingerprinting of IoT devices (hereinafter, *IoT fingerprinting*) analyses the traffic of a device and estimates the identity of the device itself, without relying on any identifier it might include in the packets it sends. Malicious or suspicious activities can be detected by comparing this estimate with the device identity associated with the identifier reported in the traffic (e.g., its MAC address). For example, IoT fingerprinting allows security practitioners to detect unknown malicious devices that are spoofing the MAC or IP address of legit devices to avoid suspicion. Additionally, authorised devices infected by a malware might start unusual network activities, which can prevent an IoT fingerprinting mechanism from recognising these devices and, consequently, enable security practitioners to identify them as suspicious.

IoT fingerprinting commonly involves passively analysing network traffic, without any direct interaction with the devices, which could otherwise allow an attacker to understand that their behaviour is being analysed. Most IoT fingerprinting approaches employ machine learning (ML) techniques to train models to recognise known IoT devices based on features extracted from network traffic. Several studies have shown the effectiveness and accuracy of passive IoT fingerprinting based on ML [24, 33, 35]. Most of these studies rely on supervised learning [1–4, 6, 7, 9, 14, 15, 23, 26, 28, 30, 31], where a classifier is trained to recognise a set of IoT devices using a dedicated class for each device. Other ML approaches used for IoT fingerprinting are based on semi-supervised [5, 34, 35, 39] and unsupervised [10, 32] techniques.

An important trend in the IoT field is the staggering and relentless increase of active IoT devices [36]. When an ML-based IoT fingerprinting mechanism is used, the classifier should be retrained often to ensure new devices can be recognised accurately. However, this aspect is largely neglected in literature. In fact, most of the IoT fingerprinting approaches proposed in literature are designed and evaluated to handle a fixed number of devices, without assessing their behaviour when new devices are introduced [11], i.e., without assessing their *scalability*. Marchal *et al.* [21] define the scalability of IoT fingerprinting as the ability "to manage a large number of IoT device types and learn to identify new types as they emerge".

A naive approach would be to retain all the available training data, augment it whenever training data for new IoT devices is available and, finally, retrain the classifier from scratch. Although sound in principle, this approach is impractical because it requires a large, growing amount of training data; also, training a model from scratch on all the training data can be very time consuming.

A few approaches for scalable IoT fingerprinting have been proposed in literature. Some use *a binary classifier for each device* and introduce further binary classifiers as training data for new IoT devices becomes available. This approach requires an ever growing number of classifiers, which might lead to memory exhaustion. Also, additional mechanisms are needed to break ties when more classifiers return a positive outcome for the same device. Furthermore, the accuracy of this approach tends to degrade over time because existing classifiers are never retrained to ensure they can properly distinguish the devices introduced latest. Other approaches are based on *Class Incremental Learning* (CIL), where the classifier is retrained using both data for new IoT devices and a selection of data for known devices. As time goes by, the selection of data for known devices needs to be capped in size for memory/disk constraints, which implies fewer samples for each device can be retained and, therefore, the classifier accuracy tends to worsen for older devices; this phenomenon is commonly referred to as catastrophic forgetting.

This paper aims to address the limitations of existing approaches for scalable IoT fingerprinting. We propose to make use of *online stream learning* (OSL) techniques [12, 13] to minimise the overhead of retraining the classifier in terms of (i) storage to retain previous training data and (ii) retraining time. This improves scalability as it makes it easier to fingerprint larger amounts of IoT devices. An OSL classifier can be updated on the fly by providing new training data; previous training data does not need to be retained and replayed, which eliminates storage requirements. The internal model of an OSL classifier is automatically extended to adapt to the new training samples, and each training sample only needs to be processed once to be learned, which helps decrease the retraining time significantly.

While OSL-based IoT fingerprinting inherently supports scalability, its practicality depends on the accuracy it can provide in recognising IoT devices. Our focus is, thus, on assessing the OSL classifier accuracy rather than its retraining overhead. Therefore, the research question we explore in this work is: *how does OSL-based IoT fingerprinting compare with other scalable approaches in terms of device recognition accuracy?*

This paper introduces *ScaNeF-IoT*, a novel approach for scalable network IoT fingerprinting. Besides relying on an OSL-based classifier to achieve scalability, it uses fixed-size traffic session payloads as features to feed the classifier. This type of features allows to cap the time required to fingerprint a session, which, on average, enables faster detection of unusual network activities.

We implement ScaNeF-IoT using the Adaptive Mondrian Forest (AMF) [25] as OSL algorithm and focusing on UDP/TCP sessions. We evaluate our implementation in terms of IoT device recognition accuracy using the IoT Traces dataset [31] and comparing the results with two state of the art scalable IoT fingerprinting approaches, namely AutoIoT [11] and IoT-Portrait [38]. The authors of these works evaluated their approaches using the same dataset; to enable a fair comparison, our experiments employ the same evaluation strategy as theirs. The former is compared with over the whole dataset, with ScaNeF-IoT showing comparable device recognition accuracy. The comparison with the latter, instead, assesses the accuracy across a number of retrainings, demonstrating that ScaNeF-IoT performs better than IoT-Portrait most of the time.

The novelty of this paper lies in the fact that ScaNeF-IoT is the first approach proposed for scalable IoT fingerprinting that is based on OSL. The main contributions of this work are:

- a novel approach for scalable IoT fingerprinting, based on OSL and fixed-size session payloads;
- an implementation of the approach, based on AMF and UDP/TCP payloads;
- an experimental evaluation based on the IoT Traces dataset [31], with an accuracy comparison with two other scalable IoT fingerprinting approaches, AutoIoT [11] and IoT-Portrait [38], showing comparable results with the former and outperforming the latter.

The rest of the paper is organised as follows. Section 2 discusses the research works carried out in the field. Section 3 presents the ScaNeF-IoT approach. Section 4 presents the experimental evaluation. Finally, conclusions and future works are presented in Section 5.

2 RELATED WORK

This section discusses academic works that propose approaches for scalable IoT fingerprinting.

Miettinen *et al.* [24] are the first to propose a scalable IoT fingerprint mechanism, called IoT Sentinel. A binary random forest classifier is trained for each considered device type; then, if there are multiple matches from different classifiers, ties are broken when using the edit distance. They use Damerau-Levenshtein [8] distance as a metric to compare fingerprints obtained from sequences of packets and select the device type with the closest match based on the computed dissimilarity scores. Ma *et al.* [19, 20] employ a similar approach to fingerprint IoT devices behind Network Address Translation (NAT). They use a binary convolutional neural network (CNN) for each IoT device. Using a binary classifier for each IoT device type is impractical, because the number of required classifiers keeps increasing linearly as new IoT devices are introduced.

Fan *et al.* [11] propose AutoIoT, which identifies new IoT devices using a few labelled samples based on multi-task learning. The traffic features undergo a preprocessing phase for feature reduction via principal component analysis (PCA) to feed the classifier. The classifier consists of a CNN with two fully connected layers, one for distinguishing between IoT and non-IoT devices, and the second for determining new device types. At the device type identification task, the distribution of the incoming features is compared against the known distribution to detect the presence of new types using the Kolmogorov-Smirnov test. The new data is transferred into low-dimension features and clustered using K-means and elbow methods to determine the optimal number of clusters, which are then labelled automatically. Finally, the model is updated by extending the fully connected layer to include the new types of devices via transfer learning. AutoIoT is evaluated using the IoT Traces dataset [31] and achieves over 95% accuracy in the identification of IoT/non-IoT devices. While the transfer learning for updating the

model shows over 99% overall accuracy, the method provides lower performance when using the YourThings dataset [29]; over 95% with the identification of IoT/non-IoT devices, over 89% with the identification of new device types, and less than 99% with transfer learning. However, The study does not evaluate experimentally how the approach performs when new IoT devices are introduced; rather, it tests the transfer learning by setting some devices as unknown and then updating the model with them.

Another approach for scalable IoT fingerprinting is based on Class incremental learning (CIL) with knowledge replay. It splits the problem incrementally into sequential tasks, where in each task a classifier is trained to recognise new devices (i.e., classes) as well as old devices [22, 37]. This is implemented by reusing samples from previous tasks, called exemplars [22, 37]. While this method can prevent catastrophic forgetting (i.e., a reduction in recognition accuracy for older classes) because the model is retrained with older devices too, it can also lead to memory exhaustion as more and more new classes are introduced over time [18]. Wang *et al.* [38] propose IoT-Portrait, an IoT fingerprinting mechanism that relies on a transformer network and a multi-classifier. They use active scanning for automatic labelling and passive scanning for fingerprinting. The transformer network is used for capturing significant network behaviour characteristics. After extracting features using the transformer, the classification model is trained using the corresponding labels. To avoid catastrophic forgetting when new devices join the network, the authors use a fixed-memory size to hold both old and new classes. The results show 85% F1 score with a fixed-memory size of 10,000 exemplars of 15 IoT devices using the IoT Traces dataset [31].

3 SCANEF-IOT APPROACH AND IMPLEMENTATION

This section introduces the ScaNeF-IoT approach for scalable network fingerprinting of IoT devices. Figure 1 highlights the key stages of the proposed approach. Network traffic generated by IoT devices is first preprocessed to (i) identify sessions, (ii) extract their payload and (iii) produce fixed-size payloads. Section 3.1 details the operations involved in this step. The resulting payloads are used as feature vectors to feed an OSL classifier, which outputs the identifier of the corresponding IoT device. More details on this step and how it enables scalability are discussed in Section 3.2. The implementation of the ScaNeF-IoT approach is described in Section 3.3.

3.1 Preprocessing

In this work, we adopt an approach similar to Kotak and Elovici [16], where images are generated from session payloads. We use packet headers to identify bi-directional sessions. The payloads of a session are concatenated and either trimmed or padded to a fixed size to generate the feature vector corresponding to that session. Rather than creating an image, we produce a fixed-size payload ready to be fed to the chosen classifier.

Using fixed-size payloads allows to generate feature vectors without waiting for the end of longer sessions, which, on average, can permit to classify a session earlier. This better fits situations where it is fundamental to take security countermeasures promptly in case anomalies are detected. While, in this way, a possibly large part of the payload might not be considered for the classification, other works in the literature have shown that considering only the initial segment of a network communication is sufficient to accurately fingerprint IoT devices [16].

The preprocessing stage consists of three steps, as shown in Figure 2.

(1) **Session identification**. By network session, we refer to the traffic exchanged between two different endpoints using the same protocol. For example, in TCP or UDP, a session is uniquely identified by the following 5 values included in each packet header: source IP address, source port number, destination IP address, destination port number and protocol (TCP or UDP).
(2) **Session payload extraction**. The payloads of the packets within a same session are extracted and concatenated in a single, session-specific buffer in hexadecimal format. Empty payload sessions are ignored.
(3) **Fixed-size payload generation**. Once the buffer reaches a size equal to or greater than a prefixed value S, the payload is trimmed to S and becomes ready for classification. In case the session ends before the buffer reaches a size of S, then it is padded with 0x00 bytes.

A fixed-size payload extracted from a session initiated by an IoT device x (i.e., with x as source of the first packet of the session) is used to train the classifier model to recognise x. Note that payloads might be encrypted.

3.2 Classification

The classification stage relies on an Online stream learning (OSL) algorithm to achieve scalability. Indeed, OSL techniques allow for incremental updates of the model as new training data is available, without the need for retraining the model from scratch every time [13] and, therefore, for storing training data for later retraining. This approach also helps address the problem of concept drift, where the distribution of the data changes over time and deviates from the one learned by the ML model [13]. Indeed, these algorithms typically employ adaptive mechanisms to adjust to changes in the data distribution over time, making them suitable for dynamic environments such as IoT networks.

When up-to-date labelled data becomes available pertaining to new IoT devices or more recent behaviour of known IoT devices, this data can be used to incrementally retrain the OSL model to enable recognising those new devices or behaviours. Contrarily to the IoT fingerprinting scalability approaches detailed in Section 2, which require keeping the whole or a selection of the training data used previously, ScaNeF-IoT relies on the adaptability properties of OSL to enable updating the classifier model using the new data only. This allows to scale the model seamlessly to recognise new IoT devices once the corresponding training data becomes available.

3.3 Implementation

To run the experimental evaluation detailed in Section 4, we have implemented the ScaNeF-IoT approach as described below.

3.3.1 Preprocessing implementation. We identify UDP and TCP sessions using the 'session' option in the SplitCap [27] tool, which

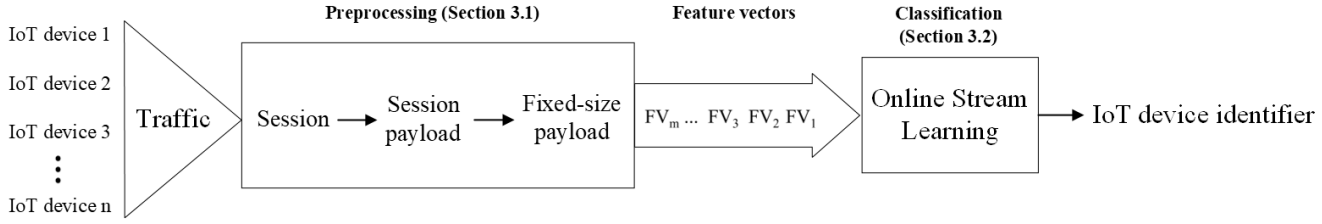

Figure 1: ScaNeF-IoT approach.

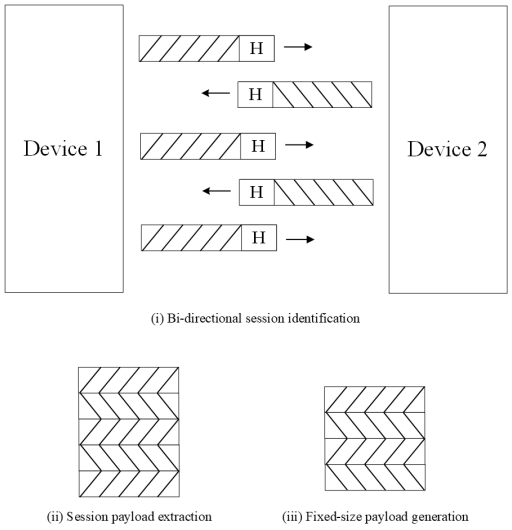

Figure 2: Preprocessing steps.

allows splitting network traffic in different sessions. We tested different values for the buffer size S to assess how it affects the accuracy of the classifier [16].

3.3.2 Classification implementation. We choose the Aggregated Mondrian Forests (AMF) as the OSL algorithm for the classification stage. AMF is an online learning algorithm for random forests that can efficiently update the model as new labelled data arrives sequentially in a streaming setting [25]. It builds upon the Mondrian Forest (MF) methodology introduced by Lakshminarayanan *et al.* [17] and incorporates principles from the Infinite Mondrian Process (IMP) proposed by Mourtada *et al.* [25] to deliver AMF.

When AMF classifies a sample, it utilises an ensemble of decision trees (DT) generated through the MP. Each DT in the ensemble provides a prediction for the sample based on its features. These individual predictions are then aggregated to produce a final prediction for the sample.

During training, When a new sample is presented for training, AMF updates the structure of its DTs to incorporate information from the sample. This may involve splitting nodes, creating new leaf nodes, or adjusting existing nodes based on the sample's features. After updating the DT structure, AMF computes a prediction for the sample using each pruned tree. These individual predictions are then aggregated using a variant of the context tree weighting (CTW) algorithm. This aggregation process assigns weights to each prediction based on its reliability, considering factors such as the tree's performance on past samples. The final prediction for the sample is obtained by computing a weighted average of all the predictions from the pruned trees.

By updating its DT structure and prediction function in an online streaming setting, AMF adapts and learns from the data incrementally, without requiring full retraining from scratch. Consequently, enabling it to make accurate predictions even in the presence of changing or evolving patterns in the data.

4 EVALUATION

This section describes how we evaluate ScaNeF-IoT to assess its accuracy in fingerprinting IoT devices, as well as its scalability as the classifier model learns new IoT devices.

Dataset. We use the IoT Traces dataset [31], which consists of network traffic traces captured in a network with 23 IoT devices and 7 non-IoT devices (e.g., smartphones, tablets, and laptops). The dataset is a collection of 20 pcap files of approximately 9.5 GB total size, spanning over 20 days. It includes 148788 sessions (97945 for IoT devices, 50843 for non-IoT devices). In this research, we only focus on IoT device sessions. Table 1 provides further details on the IoT devices in the dataset and how many TCP and UDP sessions are available for each.

Evaluation approach. Our experiments aim to assess ScaNeF-IoT accuracy in fingerprinting IoT devices through comparisons with relevant related works. We first evaluate the ScaNeF-IoT accuracy over the whole dataset for different values of the buffer size S, and compare it with the performance AutoIoT is reported to have on the very same dataset [11] (see Section 4.1). Then, we analyse how the ScaNeF-IoT accuracy varies as we incrementally train the classifier model with new IoT devices (see Section 4.2); in this experiment, the comparison is made with IoT-Portrait [38], which employs an incremental learning approach and provides an experimental evaluation based on the same dataset we use.

To enable a fair comparison with alternative approaches proposed by other researchers, we employ the same accuracy metrics they use in their experiments. In particular, we consider the overall *accuracy* of a classifier, defined as:

$$accuracy = \frac{number\ of\ correct\ classifications}{total\ number\ of\ classifications} \tag{1}$$

To assess the accuracy of the classifier in fingerprinting a specific IoT device x, we use the F1 Score $F1_x$ defined as:

$$F1_x = 2 \times \frac{Precision_x \times Recall_x}{Precision_x + Recall_x} \tag{2}$$

Table 1: The Payload Sessions Extracted from the IoT Traces Dataset

#	IoT Device Name	Sessions	TCP Sessions	UDP Sessions
1	Amazon-Echo	3491	3407	84
2	Belkin-Wemo-Motion-Sensor	48883	48786	97
3	Belkin-Wemo-Switch	8939	7032	1907
4	Blipcare-Blood-Pressure-Meter	4	4	0
5	Dropcam	35	35	0
6	HP-Printer	150	150	0
7	iHome-Power-Plug	153	153	0
8	Insteon-Camera-wired	8702	4055	4647
9	Insteon-Camera-wireless	102	1	101
10	Light-Bulbs-LiFX-Smart-Bulb	52	34	18
11	Nest-Dropcam	29	29	0
12	NEST Protect Smoke Alarm	84	84	0
13	Netatmo Weather Station	2338	2338	0
14	Netatmo Welcome	2728	2688	0
15	PIX-STAR Photo-Frame	1118	1118	0
16	Samsung-SmartCam	10053	9082	971
17	Samsung-Smart-Things	24	24	0
18	TP-Link-Day-Night-Cloud-Camera	1541	1109	432
19	TP-Link-Smart-Plug	239	232	7
20	Triby-Speaker	131	129	2
21	Withings-Aura-Smart-Sleep-Sensor	3584	3584	0
22	Withings-Smart-Baby-Monitor	5545	5545	0
23	Withings-Smart-Scale	20	20	0

where $Precision_x = \frac{TP_x}{TP_x+FP_x}$ and $Recall_x = \frac{TP_x}{TP_x+FN_x}$, with TP_x, FP_x, TN_x and FN_x defined as:

- TP_x: number of feature vectors corresponding to device x (i.e., generated from sessions initiated by x) that are classified as x;
- FP_x: number of feature vectors corresponding to a device other than x that are classified as x;
- TN_x: number of feature vectors corresponding to a device other than x that are not classified as x;
- FN_x: number of feature vectors corresponding to device x that are not classified as x.

Furthermore, we ensure using the same set of devices and the same validation strategy they employ.

4.1 Accuracy evaluation

In this experiment, we assess how the accuracy of ScaNeF-IoT , as defined in Equation 1, is affected by the buffer size. We test three different buffer sizes: 265, 784 and 1024 bytes. Additionally, we make a comparison with the classification performances reported by Fan *et al.* for AutoIoT [11], which is described in Section 2. Like them, we use all 23 IoT devices included in the IoT Traces dataset and adopt a 30% holdout validation strategy, splitting the dataset into training and test sets in a 7:3 ratio. We consider a sequential split according to the timestamp by preserving the last 30% of each IoT device payload in the test set.

Figure 3 shows the results of this experiment. While ScaNeF-IoT always showcases an accuracy comparable to AutoIoT, a buffer size of 784 bytes leads to a slightly higher accuracy overall (99.29% with 784 bytes against 99.18% with 265 and 99.24% with 1024).

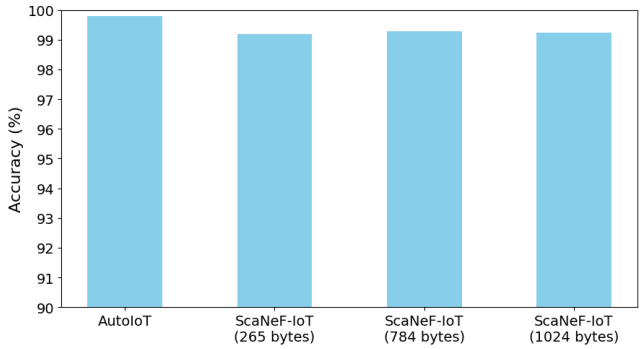

Figure 3: Accuracy comparison between AutoIoT and ScaNeF-IoT with different buffer sizes, based on a 30% holdout validation strategy.

However, since the difference in accuracy is very small, we look at the F1 score of each device, as defined in Equation 2, to analyse the extent to which different buffer sizes affect the fingerprinting of individual devices. For each buffer size, we assess how many devices are fingerprinted with the highest F1 score by ScaNeF-IoT when configured with that buffer size. We distinguish between cases where the F1 score is strictly higher than with the other two buffer sizes, and cases where it is equal to the F1 score obtained with any or both of the other two buffer sizes. The results reported in Figure 4 show that ScaNeF-IoT configured with a buffer size of 784 bytes provides the highest F1 score for 19 devices out of 23 (strictly higher than the others for 4 devices, equal to the others for 15 devices). Instead, ScaNeF-IoT configured with a buffer size of 1024 bytes achieves the highest F1 score in 17 cases (strictly higher in 2 cases, equal in 15 cases), while using 265 bytes as buffer size leads to the highest F1 score for 14 devices (for 1 strictly higher, for 12 as high as the others). Therefore, for the following experiments, we configure ScaNeF-IoT with a buffer size of 784 bytes.

Although the F1 score is at least 90% for the large majority of IoT devices, ScaNeF-IoT exhibits a less than satisfactory performance for a few devices. Table 2 details precision, recall and F1 score obtained for each device when using ScaNeF-IoT configured with a buffer size of 784 bytes. Also, the table reports the support for each device, which is the number of feature vectors used for the testing. The dataset is heavily imbalanced, as evidenced by the widely varying support values across devices, which contributes to the lower F1 scores observed in some devices.

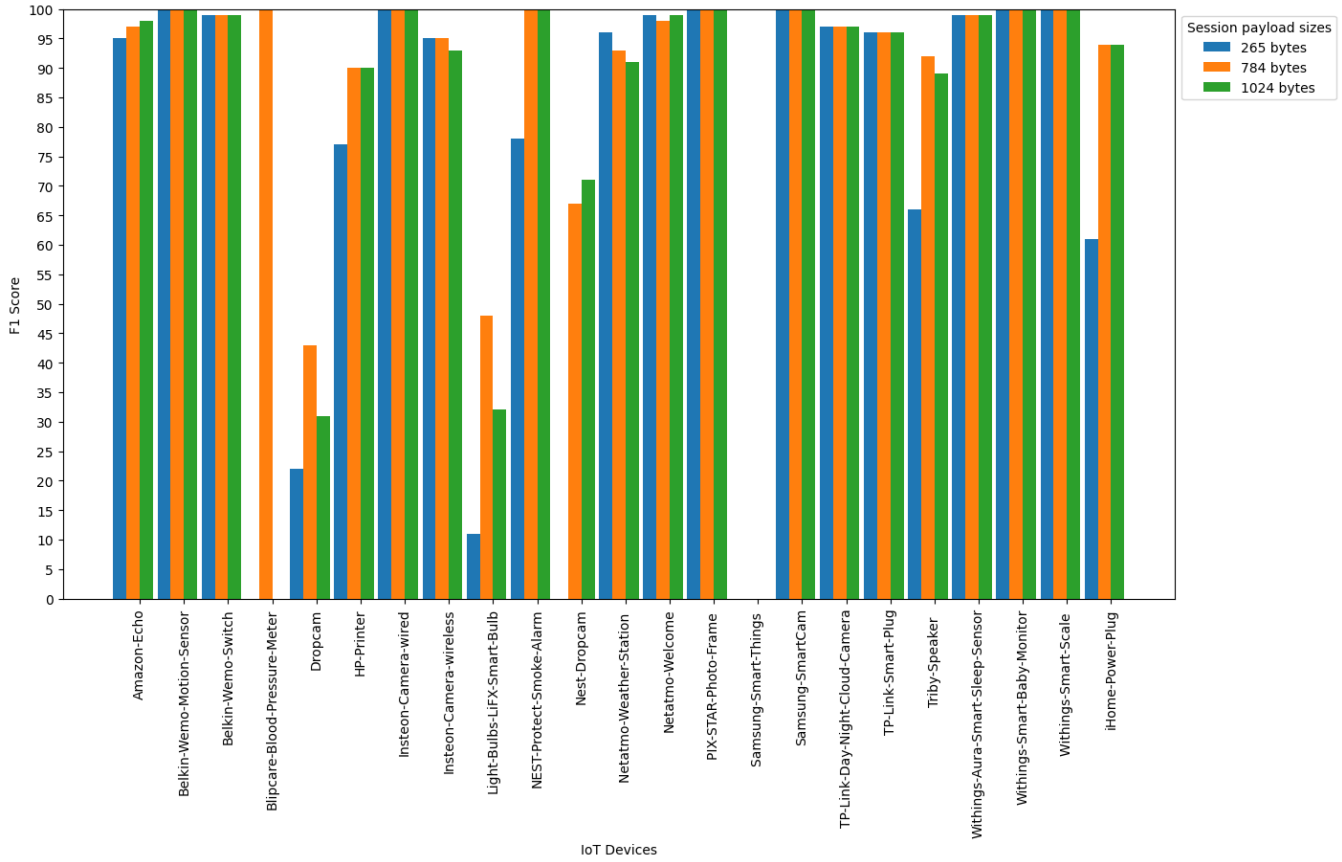

Figure 4: F1 score comparison for each IoT device when using ScaNeF-IoT with different buffer sizes.

4.2 Scalability evaluation

This experiment aims to assess how ScaNeF-IoT fingerprinting accuracy varies as the classifier model is updated to recognise new devices. We compare ScaNeF-IoT's performance with a state of the art scalable IoT fingerprinting approach, namely IoT-Portrait [38], which is described in Section 2.

To ensure fairness in the comparison, we scale up the model in the same way IoT-Portrait does. As explained in Section 2, IoT-Portrait employs CIL and, therefore, arranges its classification activities over time across a number of sequential tasks. At the beginning of each task, a set of new IoT devices is introduced and the classifier model is retrained to recognise these new devices as well as all the IoT devices introduced in the previous tasks; in the first task (i.e., task 0), the classifier model is trained to fingerprint an initial set of IoT devices.

IoT-Portrait is evaluated using the IoT Traces dataset with 3 IoT devices in task 0 and 2 new IoT devices in each of the following 6 tasks. Table 3 details which IoT devices are new in each task. To assess ScaNeF-IoT, we update the classifier model using the same sets of IoT devices that Wang *et al.* used to evaluate IoT-Portrait. Some of these IoT devices are under-represented in the dataset, which can lead to poor accuracy, as discussed in Section 4.1. Also, we adopt a 6:4 ratio for training and testing since the same is used to assess IoT-Portrait. The test set for each task includes payloads from new devices introduced and payloads from older devices presented in previous tasks. The accuracy is calculated for each task as defined in Equation 1, and is reported in Table 3.

Figure 5 shows the comparison between IoT-Portrait and ScaNeF-IoT in the accuracy they provide across all 7 tasks. ScaNeF-IoT performs better than or equally to IoT-Portrait in all tasks but task 3, where the accuracy drops to 71%.

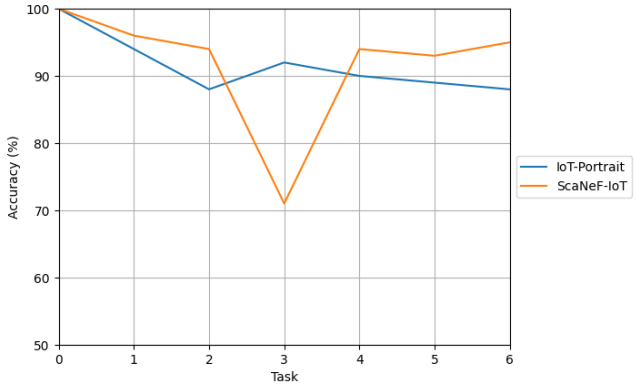

Figure 5: Accuracy comparison between ScaNeF-IoT and IoT-Portrait as new IoT devices are introduced.

Table 2: Precision, recall, F1 score and support for each IoT device using ScaNeF-IoT with 784 bytes buffer size.

#	IoT Device Name	Precision	Recall	F1	Support
1	Amazon-Echo	0.99	0.96	0.97	1047
2	Belkin-Wemo-Motion-Sensor	1.00	1.00	1.00	14665
3	Belkin-Wemo-Switch	1.00	0.99	0.99	2682
4	Blipcare-Blood-Pressure-Meter	1.00	1.00	1.00	1
5	Dropcam	1.00	0.27	0.43	11
6	HP-Printer	1.00	0.82	0.90	45
7	Insteon-Camera-wired	1.00	1.00	1.00	2611
8	Insteon-Camera-wireless	1.00	0.90	0.95	31
9	Light-Bulbs-LiFX-Smart-Bulb	1.00	0.31	0.48	16
10	NEST-Protect-Smoke-Alarm	1.00	1.00	1.00	25
11	Nest-Dropcam	0.83	0.56	0.67	9
12	Netatmo-Weather-Station	0.87	1.00	0.93	701
13	Netatmo-Welcome	0.99	0.98	0.98	818
14	PIX-STAR-Photo-Frame	1.00	1.00	1.00	335
15	Samsung-Smart-Things	0.00	0.00	0.00	7
16	Samsung-SmartCam	1.00	1.00	1.00	3016
17	TP-Link-Day-Night-Cloud-Camera	0.98	0.97	0.97	462
18	TP-Link-Smart-Plug	1.00	0.93	0.96	72
19	Triby-Speaker	1.00	0.85	0.92	39
20	Withings-Aura-Smart-Sleep-Sensor	0.99	0.99	0.99	1075
21	Withings-Smart-Baby-Monitor	0.99	1.00	1.00	1664
22	Withings-Smart-Scale	1.00	1.00	1.00	6
23	iHome-Power-Plug	1.00	0.89	0.94	46

Table 3: Fingerprinting accuracy of ScaNeF-IoT as the classifier model is scaled up to recognise new IoT devices.

#	Device Name	Task	Accuracy
1	Samsung-SmartCam	0	99.97
2	Belkin-Wemo-Motion-Sensor		
3	Withings-Smart-Baby-Monitor		
4	Belkin-Wemo-Switch	1	96.11
5	Amazon-Echo		
6	Netatmo-Welcome	2	94.33
7	Netatmo-Weather-Station		
8	TP-Link-Day-Night-Cloud-Camera	3	71.15
9	PIX-STAR-Photo-Frame		
10	TP-Link-Smart-Plug	4	94.14
11	HP-Printer		
12	Triby-Speaker	5	93.47
13	Dropcam		
14	Samsung-Smart-Things	6	94.74
15	Withings-Smart-Scale		

This outlier is mostly caused by the misclassification of a large proportion of Belkin-Wemo-Motion-Sensor samples as TP-Link-Day-Night-Cloud-Camera, as reported in the task 3 confusion matrix in Figure 6. The F1 score for these two devices in task 3 is 73% and 12%, respectively.

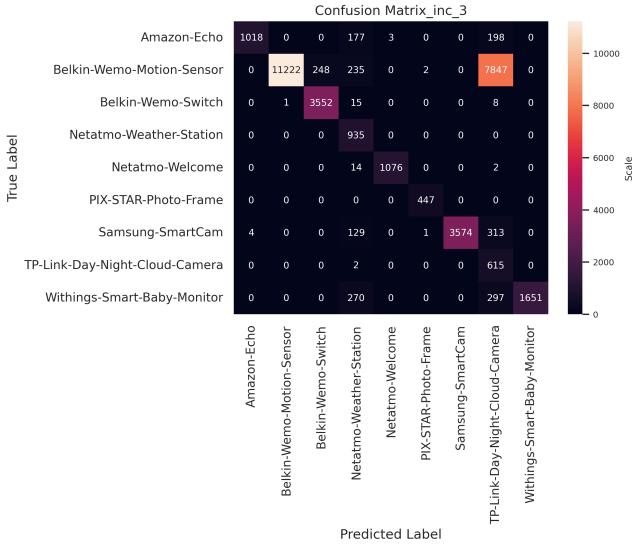

Figure 6: Task 3 confusion matrix

In the following task, once the classifier model has been updated to recognise the two new IoT devices, the overall accuracy goes back to 94%. Also, the misclassifications of Belkin-Wemo-Motion-Sensor and TP-Link-Day-Night-Cloud-Camera samples are reduced significantly, as shown in Figure 7, with F1 score for these two devices equal to 98% and 73%, respectively.

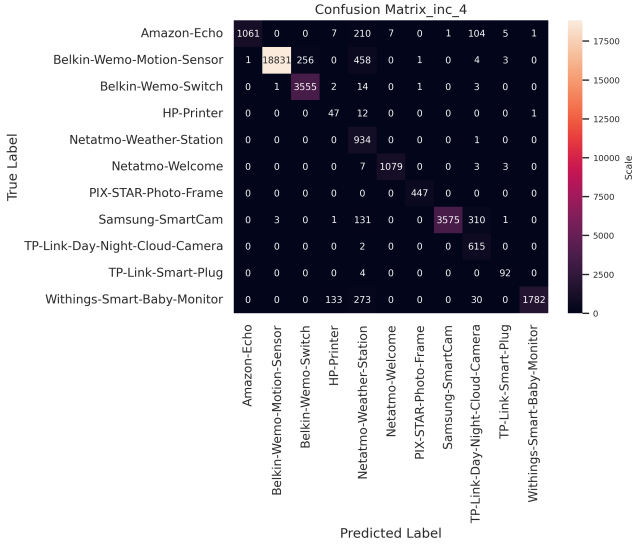

Figure 7: Task 4 confusion matrix

5 CONCLUSION

Scalable approaches to IoT fingerprinting are required to cope with the fast pace at which new IoT devices are developed. The novel approach we propose, ScaNeF-IoT, achieves scalability by using an OSL classifier to reduce storage requirements and retraining time. The research question we address in this paper is whether ScaNeF-IoT can offer a device recognition accuracy comparable to that provided by existing scalable IoT fingerprinting approaches, namely AutoIoT and IoT-Portrait. Our experiments show that ScaNeF-IoT performs similarly to AutoIoT when assessed on all the IoT devices included in a same dataset, and outperforms IoT-Portrait when incrementally introducing new IoT devices. While this is a preliminary evaluation, the results are promising towards establishing OSL as the ML classifier to use to achieve scalable IoT fingerprinting.

This line of research can be developed further by assessing and comparing the retraining time against existing approaches. Also, other OSL algorithms beyond AMF can be tested, as well as combining different types of features (e.g., statistics of the sessions) to explore the possibility to improve the detection accuracy even further.

ACKNOWLEDGMENTS

This work was partially supported by the Academic Centre of Excellence in Cyber Security Research - University of Southampton (EP/R007268/1). The authors gratefully acknowledge the generous financial support from Imam Mohammad Ibn Saud Islamic University (IMSIU) via the Saudi Arabian Cultural Bureau in London.

REFERENCES

[1] Abbas Acar, Hossein Fereidooni, Tigist Abera, Amit Kumar Sikder, Markus Miettinen, Hidayet Aksu, Mauro Conti, Ahmad-Reza Sadeghi, and Selcuk Uluagac. 2020. Peek-a-boo: i see your smart home activities, even encrypted!. In *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*.

[2] A. Aksoy and M. H. Gunes. 2019. Automated IoT Device Identification using Network Traffic. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*.

[3] Sandhya Aneja, Nagender Aneja, and Md Shohidul Islam. 2018. IoT Device Fingerprint using Deep Learning. In *2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*.

[4] Lei Bai, Lina Yao, Salil S. Kanhere, Xianzhi Wang, and Zheng Yang. 2018. Automatic Device Classification from Network Traffic Streams of Internet of Things. In *2018 IEEE 43rd Conference on Local Computer Networks (LCN)*.

[5] Jiaqi Bao, Bechir Hamdaoui, and Weng-Keen Wong. 2020. IoT Device Type Identification Using Hybrid Deep Learning Approach for Increased IoT Security. In *2020 International Wireless Communications and Mobile Computing (IWCMC)*.

[6] Bruhadeshwar Bezawada, Maalvika Bachani, Jordan Peterson, Hossein Shirazi, Indrakshi Ray, and Indrajit Ray. 2018. Behavioral Fingerprinting of IoT Devices. In *Proceedings of the 2018 Workshop on Attacks and Solutions in Hardware Security*.

[7] Joe Collins, Michaela Iorga, Dmitry Cousin, and David Chapman. 2020. Passive Encrypted IoT Device Fingerprinting with Persistent Homology. In *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond*.

[8] Fred J. Damerau. 1964. A technique for computer detection and correction of spelling errors. *Commun. ACM* (1964).

[9] B. A. Desai, D. M. Divakaran, I. Nevat, G. W. Peter, and M. Gurusamy. 2019. A feature-ranking framework for IoT device classification. In *2019 11th International Conference on Communication Systems Networks (COMSNETS)*.

[10] Ruizhong Du and Shuang Li. 2021. Identification of IoT Devices Based on Feature Vector Split. In *2021 IEEE Symposium on Computers and Communications (ISCC)*.

[11] Linna Fan, Lin He, Yichao Wu, Shize Zhang, Zhiliang Wang, Jia Li, Jiahai Yang, Chaocan Xiang, and Xiaoqian Ma. 2022. AutoIoT: Automatically Updated IoT Device Identification With Semi-Supervised Learning. *IEEE Transactions on Mobile Computing* (2022).

[12] João Gama, Raquel Sebastião, and Pedro Pereira Rodrigues. 2013. On evaluating stream learning algorithms. *Mach. Learn.* (2013).

[13] Heitor Murilo Gomes, Jesse Read, Albert Bifet, Jean Paul Barddal, and João Gama. 2019. Machine learning for streaming data: state of the art, challenges, and opportunities. *SIGKDD Explor. Newsl.* (2019).

[14] S. A. Hamad, W. E. Zhang, Q. Z. Sheng, and S. Nepal. 2019. IoT Device Identification via Network-Flow Based Fingerprinting and Learning. In *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*.

[15] Jaidip Kotak and Yuval Elovici. 2021. IoT Device Identification Using Deep Learning. In *13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020)*.

[16] Jaidip Kotak and Yuval Elovici. 2022. IoT device identification based on network communication analysis using deep learning. *Journal of Ambient Intelligence and Humanized Computing* (2022).

[17] Balaji Lakshminarayanan, Daniel M Roy, and Yee Whye Teh. 2014. Mondrian forests: Efficient online random forests. *Advances in neural information processing systems* (2014).

[18] Yongxin Liu, Jian Wang, Jianqiang Li, Shuteng Niu, and Houbing Song. 2021. Class-Incremental Learning for Wireless Device Identification in IoT. *IEEE Internet of Things Journal* (2021).

[19] Xiaobo Ma, Jian Qu, Jianfeng Li, John C.S. Lui, Zhenhua Li, and Xiaohong Guan. 2020. Pinpointing Hidden IoT Devices via Spatial-temporal Traffic Fingerprinting. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*.

[20] Xiaobo Ma, Jian Qu, Jianfeng Li, John C. S. Lui, Zhenhua Li, Wenmao Liu, and Xiaohong Guan. 2022. Inferring Hidden IoT Devices and User Interactions via Spatial-Temporal Traffic Fingerprinting. *IEEE/ACM Transactions on Networking* (2022).

[21] S. Marchal, M. Miettinen, T. D. Nguyen, A. Sadeghi, and N. Asokan. 2019. AuDI: Toward Autonomous IoT Device-Type Identification Using Periodic Communication. *IEEE Journal on Selected Areas in Communications* (2019).

[22] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D. Bagdanov, and Joost van de Weijer. 2023. Class-Incremental Learning: Survey and Performance Evaluation on Image Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[23] Yair Meidan, Michael Bohadana, Asaf Shabtai, Juan David Guarnizo, Martín Ochoa, Nils Ole Tippenhauer, and Yuval Elovici. 2017. ProfilIoT: A Machine Learning Approach for IoT Device Identification Based on Network Traffic Analysis. In *Proceedings of the Symposium on Applied Computing*.

[24] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A. Sadeghi, and S. Tarkoma. 2017. IoT SENTINEL: Automated Device-Type Identification for Security Enforcement in IoT. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*.

[25] Jaouad Mourtada, Stéphane Gaïffas, and Erwan Scornet. 2021. AMF: Aggregated Mondrian forests for online learning. *Journal of the Royal Statistical Society Series B: Statistical Methodology* (2021).

[26] N. Msadek, R. Soua, and T. Engel. 2019. IoT Device Fingerprinting: Machine Learning based Encrypted Traffic Analysis. In *2019 IEEE Wireless Communications and Networking Conference (WCNC)*.

[27] Netresec. 2010. *SplitCap: A Tool for Network Traffic Analysis*. https://www.netresec.com/?page=SplitCap#

[28] R. Perdisci, T. Papastergiou, O. Alrawi, and M. Antonakakis. 2020. IoTFinder: Efficient Large-Scale Identification of IoT Devices via Passive DNS Traffic Analysis. In *2020 IEEE European Symposium on Security and Privacy (EuroS P)*.

[29] Jingjing Ren, Daniel J. Dubois, David Choffnes, Anna Maria Mandalari, Roman Kolcun, and Hamed Haddadi. 2019. Information Exposure From Consumer IoT Devices: A Multidimensional, Network-Informed Measurement Approach. In *Proceedings of the Internet Measurement Conference*.

[30] M. R. Shahid, G. Blanc, Z. Zhang, and H. Debar. 2018. IoT Devices Recognition Through Network Traffic Analysis. In *2018 IEEE International Conference on Big Data (Big Data)*.

[31] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman. 2019. Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics. *IEEE Transactions on Mobile Computing* (2019).

[32] A. Sivanathan, H. H. Gharakheili, and V. Sivaraman. 2019. Inferring IoT Device Types from Network Behavior Using Unsupervised Clustering. In *2019 IEEE 44th Conference on Local Computer Networks (LCN)*.

[33] Jianhua Sun, Kun Sun, and Chris Shenefiel. 2019. Automated IoT Device Fingerprinting Through Encrypted Stream Classification. In *Security and Privacy in Communication Networks*.

[34] Yi Sun, Jie Liu, Ali Kashif Bashir, Usman Tariq, Wei Liu, Keliang Chen, and Mohammad Dahman Alshehri. 2021. E-CIS: Edge-based classifier identification scheme in green & sustainable IoT smart city. *Sustainable Cities and Society* (2021).

[35] Vijayanand Thangavelu, D. Divakaran, R. Sairam, S. Bhunia, and M. Gurusamy. 2019. DEFT: A Distributed IoT Fingerprinting Technique. *IEEE Internet of Things Journal* (2019).

[36] Lionel Sujay Vailshery. 2023. *Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2023, with forecasts from 2022 to 2030*. https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

[37] Gido M. van de Ven, Zhe Li, and Andreas S. Tolias. 2021. Class-Incremental Learning With Generative Classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[38] Juan Wang, Jing Zhong, and Jiangqi Li. 2023. IoT-Portrait: Automatically Identifying IoT Devices via Transformer with Incremental Learning. *Future Internet* (2023).

[39] L. Yu, T. Liu, Z. Zhou, Y. Zhu, Q. Liu, and J. Tan. 2018. WDMTI: Wireless Device Manufacturer and Type Identification Using Hierarchical Dirichlet Process. In *2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*.