# IOT Fingerprint generation and Identification

***A B. Tech Project Report Submitted***
***in Partial Fulfillment of the Requirements***
***for the Degree of***

**Bachelor of Technology**

*by*

**Unique Kaushik**
(1901CS66)

*under the guidance of*

**Dr. Somanath Tripathy**


**to the**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**
**INDIAN INSTITUTE OF TECHNOLOGY PATNA**
**PATNA - 800013, BIHAR**

# CERTIFICATE

*This is to certify that the work contained in this thesis entitled "**IOT Fingerprint generation and Identification**" is a bonafide work of **Unique Kaushik (Roll No. 1901CS66**), carried out in the Department of Computer Science and Engineering, Indian Institute of Technology Patna under my supervision and that it has not been submitted elsewhere for a degree.*

Supervisor: **Dr. Somanath Tripathy**

Professor,

May, 2023

Department of Computer Science & Engineering,

Patna.

Indian Institute of Technology Patna, Bihar.

# Acknowledgements

First, I want to thank the Indian Institute of Technology Patna, that I could be the part of such a prestigious institute and use its facilities for to enhance my knowledge and skills.

Then, I want to pay my gratitude to my project guide, Dr. Somanath Tripathy for his unwavering and constant support throughout the entire process. Without his able guidance, this project could not have taken place at all. He has been instrumental at every single step of composition of this entire project. It is impossible for me to imagine this project happen without his mentorship.

Secondly, I thank all the faculty members for their constructive criticism. It is because of their eager research prowess; our institute has gained such recognition. They are truly a lighthouse of knowledge.

Finally, I thank all my family members and friends for being a constant pillar of support in my life. They have been present throughout all my troublesome phases, as well as my happy moments.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

The rapid growth of the Internet of Things (IoT) has led to an exponential increase in the number of connected devices. According to the IOT analytics research 2022 ,it is expected that there will be around 27 billion linked IoT devices by 2025, as supply restrictions ease and demand increases. As the number of devices grows, so does the need to correctly identify and categorise them for a variety of purposes such as network management, security, and performance optimisation. Because of their diversified nature, large range of manufacturers, and different communication protocols, identifying IoT devices is a challenging task

Machine learning algorithms have been popular in recent years for identifying IoT devices. These methods try to learn the traits and patterns displayed by IoT devices to create accurate classification models. To train ML models, we must first transform IOT data into feature vectors. Static and dynamic fingerprinting technologies are commonly employed for this.

In Static analysis we focus on extracting features from individual packets without considering any packet which came before or after it. Features such as packet size, protocol type, and port numbers are commonly used in static fingerprinting. On the other hand, in

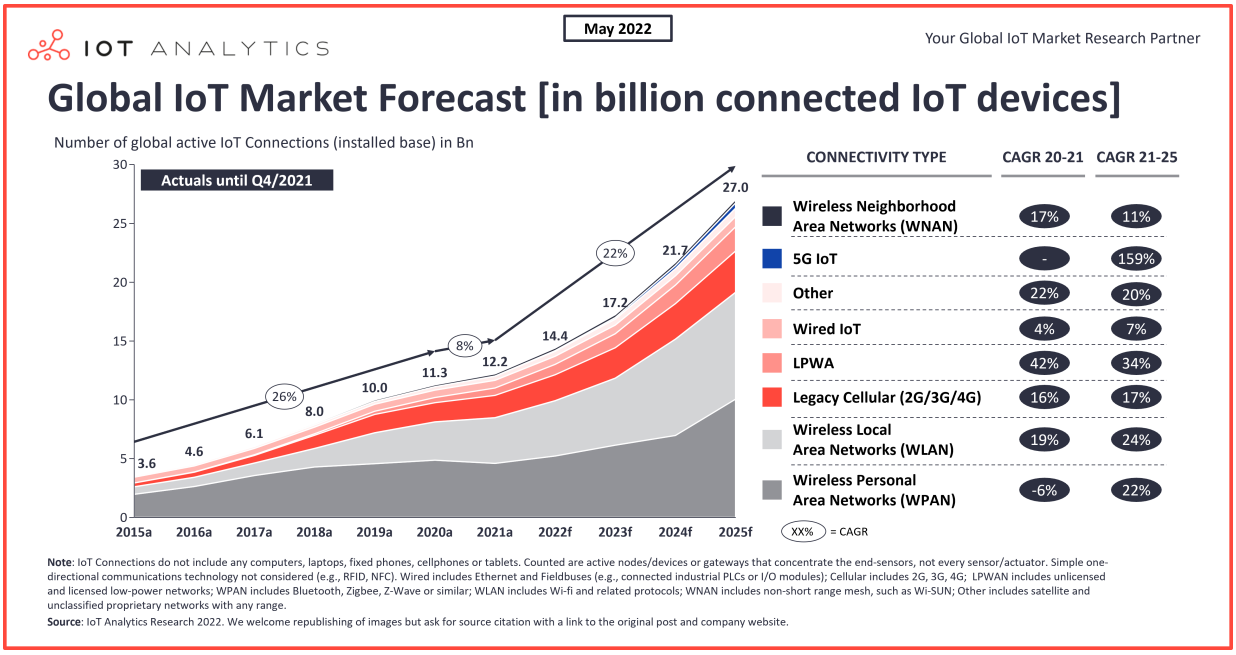



**Fig. 1.1**: Number of IOT devices by year

dynamic analysis we try to capture dynamic info by combining multiple packets or flows to form a composite fingerprint.

In this study, we aim to investigate the performance of various machine learning algorithms in the identification of IoT devices using both static and dynamic fingerprinting methods. We will evaluate the accuracy of models trained on different datasets, including the Aalto dataset and the UNSW dataset. Furthermore, we will explore the impact of additional features proposed by us, such as TTL, Direction, TCP Flags, Byte Distribution of Payload, TLS Cipher Suites, and TLS Extensions, on the classification performance of these models.

## 1.1 Organization of The Report

This section provides an overview of the topics covered in this thesis. First, we described the problem of fingerprinting and identifying IOT devices. Then we provided the approach used for accomplishing the task. The approach mainly consists of 4 parts: packet capturing, fingerprint extraction, classifier training, and prediction. The next chapter is performance

evaluation, in which we evaluated and compared the performance of different machine learning models over two datasets using various fingerprinting methods. And finally in last chapter, we conclude the thesis with some future works.

# Chapter 2

# Problem Description

Before defining the problem let us first define some terms and symbols.

Let:

$D$ be the set of all possible type of IOT devices. ($|D| \rightarrow$ number of IOT devices categories)

$X_{D_i}$ be the collection of all packets of device $D_i \in D$.

We can divide the problem into two tasks:

- Fingerprinting: Given $X_{D_i}$ the packets of device $D_i$, devise a fingerprinting function $f : X_{D_i} \rightarrow F$, where $F$ is the set of fingerprints extracted from the network traffic data for device $D_i$.

- Identification: Given a set of labeled fingerprints $S = \{(F_1, y_1), (F_2, y_2), \ldots, (F_n, y_n)\}$, where $y_i \in D$ is the true device label for $F_i$ which is one of the fingerprints of device $D_{y_i}$. We have to learn a classification function $C : F \rightarrow y$ that accurately predicts the device category of IOT device based on their fingerprints.

# Chapter 3

# Approach

In this section we give an overview of the approach and the different fingerprinting methods used.

We can divide our approach in four major components: packet capture, fingerprint extraction, model training, and Evaluation and prediction.

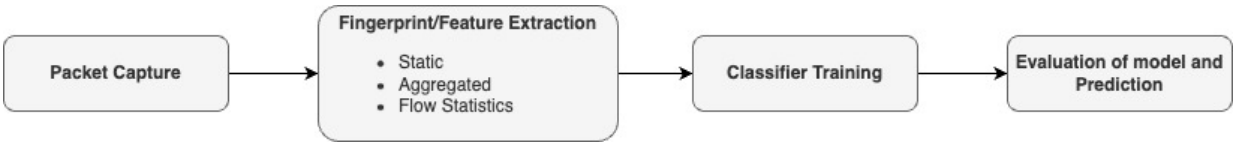



**Fig. 3.1**: Approach

## 3.1 Packet Capturing

Packet capturing, which is also known as packet sniffing or network traffic monitoring, refers to the practice of capturing and examining data packets transmitted over a network. In this technique packets are captured using any packet sniffer or tool like tshark, wireshark etc. which are flowing through a netowrk interface. However, in this thesis, we did not perform the actual packet capturing; instead, we utilized pre-captured network data described in 4.1.

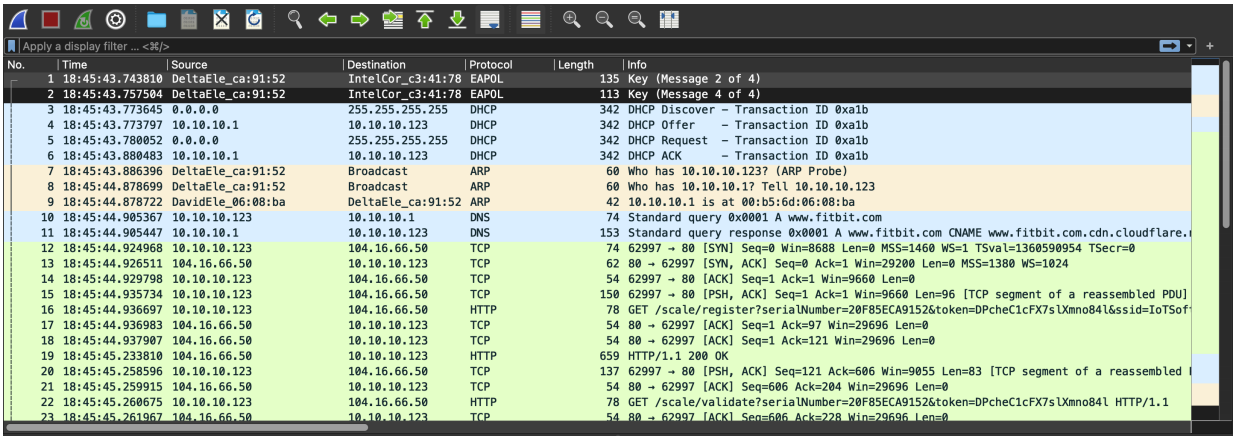



**Fig. 3.2**: Packet capture using wireshark

## 3.2 Fingerprint Extraction

The IOT Fingerprint of a device refers to the set of features or data that can be used to uniquely identify a device in the network. Fingerprint extraction or fingerprinting refers to the process of extracting these fingerprints from the network data. For extracting fingerprints, we have used three different techniques: Static fingerprinting, dynamic fingerprinting using packet aggregation, and dynamic fingerprinting using flows in the network. Now we will discuss them one by one.

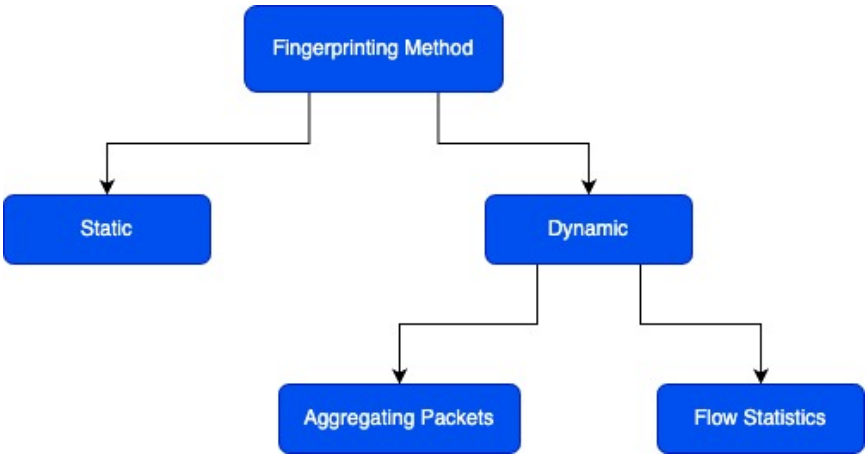



**Fig. 3.3**: Fingerprinting Methods

### 3.2.1 Static Fingerprinting

Static fingerprinting refers to a fingerprinting technique in which we extract fingerprints from individual packets without considering the preceding or subsequent packets. In this approach, we focus on extracting valuable information from the packet header, payload, and some packet statistics. The features extracted from each packet include details such as the packet's header fields, payload content, and various statistical metrics. A comprehensive list of these extracted features can be found in Tables 3.1 and 3.2. By combining these features, we can create unique fingerprints for each packet, using which we can identify and classify IoT devices based on their distinct characteristics.

| Features | Description |
|---|---|
| Protocols Used | 0/1 value if Protocols used in packet |
| Source Port (src port) | Bin index of Source port number |
| Destination Port (dst port) | Bin index of Destination port number |
| TCP Window Size (tcp w_size) | TCP window size |
| Packet Size (pkt size) | Size of the packet payload |
| Entropy of Payload | Measure of randomness in the payload |

**Table 3.1**: Features used by IOT Sentinel

| Features | Descriptiom |
|---|---|
| Byte Distribution | Distribution of byte values in the payload |
| Time-to-Live (TTL) | Remaining lifespan of the packet |
| Direction | Direction of packet transmission |
| TCP Flags | Flags set in TCP packets (e.g., SYN, ACK, FIN) |
| Cipher Suites | Cipher suites used by TLS layer |
| Extensions | Advertised Extensions in TLS layer |

**Table 3.2**: Added Features

Now we will discuss the importance of some of the features and how they can help in identifying the IOT devices:

Protocols Used: Different IoT devices uses different set of protocols to communicate based on their requirments. Due to this, we can distinguish devices based on their preferred communication mechanisms and protocols.

Source Port (src port): The source port number can be linked to certain apps or services that IoT devices utilise. We can distinguish devices depending on the programmes or services they use by analysing the source ports.

Destination Port (dst port): Similar to the source port, the destination port can provide us information about the packets's intended applications or services.

Entropy of Payload: The entropy of the payload indicates the level of randomness or information content within the packet. Entropy can help us identify which type of data is being transmitted in the packet. For ex: entropy of simple plain text data is generally lower, whereas entropy of audio/video data is generally on the higher side. So, by analyzing the entropy of the payload, we can infer about the typw the data being transmitted. For instance, if the entropy is high, it suggests that the IoT device might be a camera or another device that transmits audio or video data.

Byte Distribution: Similar to entropy, Byte distribution of the payload can reveal device-specific patterns or signatures in the payload. By analyzing byte distributions, we can distinguish devices based on their unique payload characteristics.

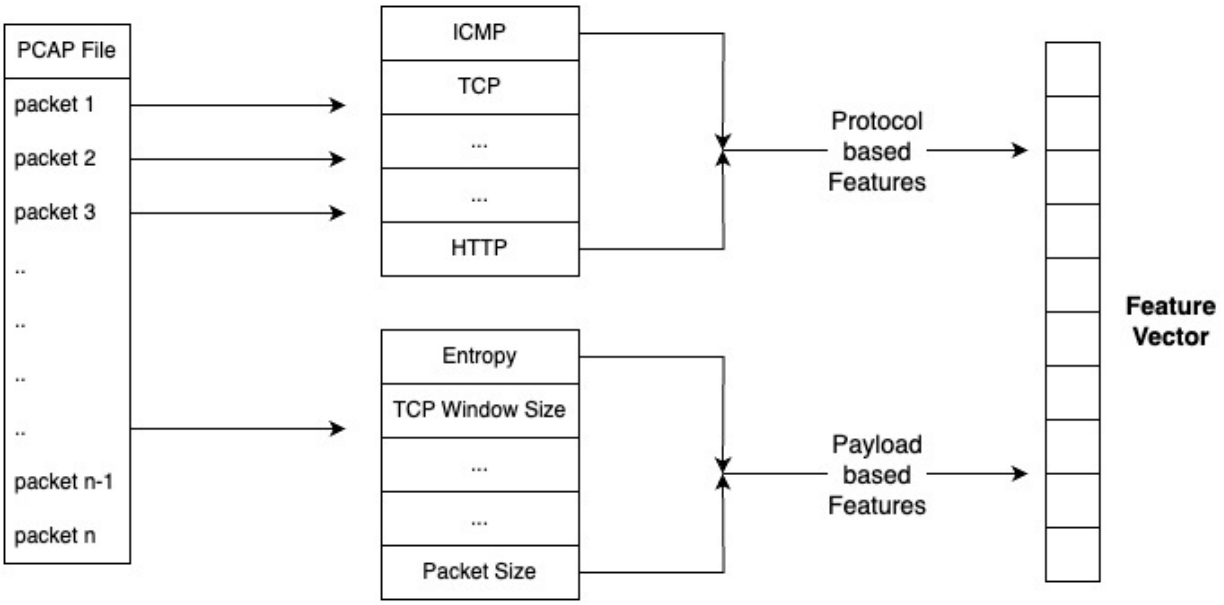



**Fig. 3.4**: Static Fingerprint Extraction

### 3.2.2 Dynamic Fingerprinting using Aggregation

Dynamic fingerprinting refers to a fingerprinting technique in which we extract single fingerprints from multiple packets by considering the preceding and subsequent packets. With dynamic fingerprinting we will be able to capture the contextual information and the communication patterns used by the device. Using dynamic fingerprints a model can learn and identify patterns that may not be learned when analyzing individual packets alone.

In packet aggregation, multiple packets are grouped together to form a single fingerprint,so using aggregation of packets we need to specify the number of packets to be aggregated for a single fingerprint. So, to find the best size of packet aggregation we perform a systematic analysis using various sizes of packet aggregation. We plotted the packet size vs the accuracy of the classifier. And the size with best acccuracy is selected for identifying the IOT devices.

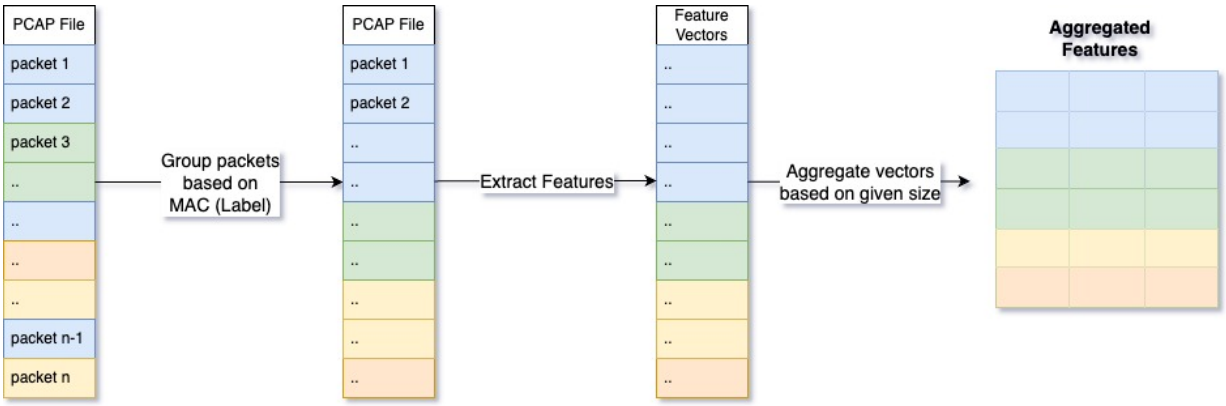



**Fig. 3.5**: Aggregated Fingerprint Extraction

Fig 3.5 shows the process of packet aggregation for dynamic fingerprinting. To perform packet aggregation, we initially group the packets in the pcap file based on the MAC address of the IoT devices. Next, we extract the relevant features from each individual packet, following a similar approach to the one used in static fingerprinting.

Once we have obtained the feature vectors for each packet, we concatenate feature vectors which are in same group based on the aggregation size to form a dynamic fingerprint.

This aggregation size determines the number of packets that will be combined to form a dynamic fingerprint. For example, if the aggregation size is set to 5, we group five consecutive packets together to create a dynamic fingerprint.

### 3.2.3 Dynamic Fingerprinting using network flows

A network flow can be described as a collection of packets that are interconnected and associated with each other. In simple terms we can say that it is the whole sequence of packets exchanged in a communication session.

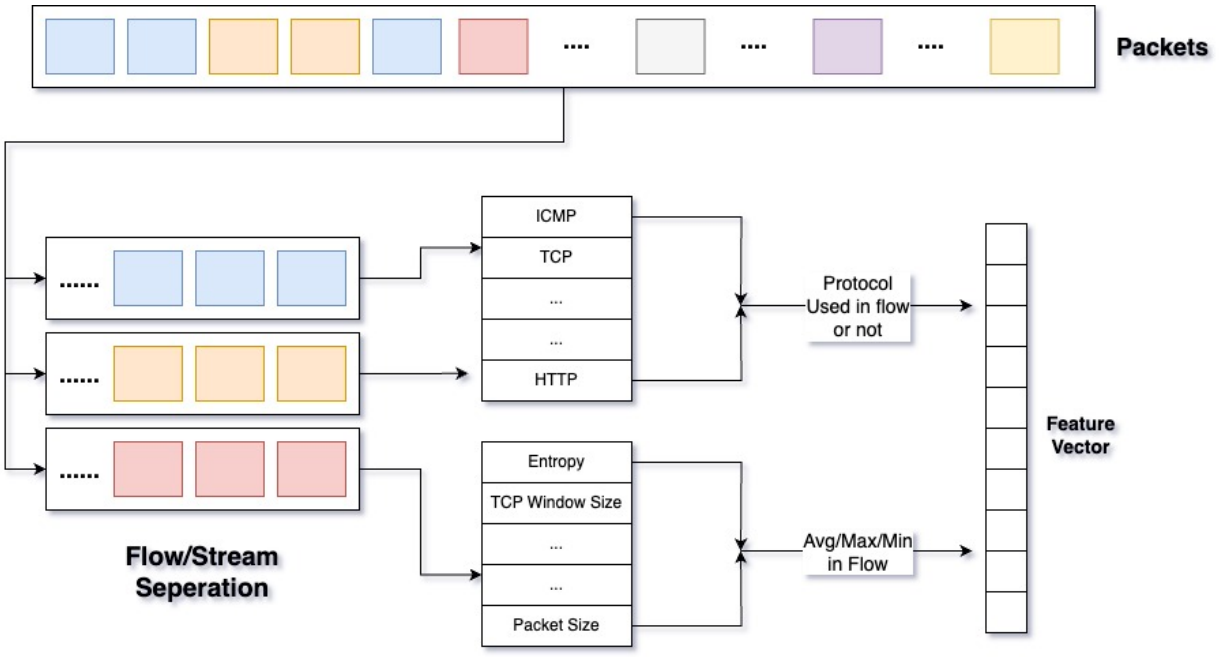



**Fig. 3.6**: Flow based Fingerprint Extraction

Similar to aggregation method, In flow-based fingerprinting, the focus is on extracting and analyzing network flows or streams, rather than individual packets. The process of extracting fingerprints from network flows involves several steps, as illustrated in Figure 3.6. First, we identify and extract the network flows or streams from the captured network traffic. Since the number of packets within each flow can vary, instead of aggregating them, we focus on extracting flow-level statistics and features. Some of the extracted features include the maximum packet size observed within the flow, the minimum packet size, the

normalized byte distribution, the presence of specific protocols within the flow etc.

## 3.3 Classifier Training

In this section, we conducted training and evaluation of various machine learning classifiers: Support Vector Machines (SVM), Naive Bayes (NB), Decision Trees (DT), and Random Forest (RF) etc. on the different types of fingerprints that were extracted in the previous section.

The aim was to determine the best classifier among all the classifiers trained for accurately identifying and classifying IoT devices based on the extracted fingerprints. We trained each classifier on the training data and evaluated their performance using appropriate evaluation metrics. Results of this are shown in section 4

## 3.4 Predicting Device Type

In this section, we focused on predicting the device type based on network data that was not included in the training phase. For the unseen data, first fingerprint was generated using the methods discussed above, and then it was fed as input to the classifier which then gave us the device type with the max probability.

The objective was to assess the performance of our trained models in accurately classifying unseen network data and identifying the device type associated with it.

# Chapter 4

# Performance Evaluation

## 4.1 Datasets Used overview

In this section, we provide an overview of the two datasets used in our study: the Aalto dataset and the UNSW dataset.

| | |
|---|---|
| HueBridge | 65959 |
| D-LinkDoorSensor | 19674 |
| D-LinkSwitch | 12950 |
| D-LinkSensor | 12691 |
| D-LinkWaterSensor | 12098 |
| D-LinkSiren | 11813 |
| WeMoLink | 10998 |
| D-LinkCam | 7474 |
| Lightify | 7421 |
| WeMoInsightSwitch | 6871 |
| WeMoSwitch | 6037 |
| EdimaxPlug1101W | 1776 |

| | |
|---|---|
| EdimaxPlug2101W | 1657 |
| EdnetGateway | 1425 |
| Withings | 1382 |
| TP-LinkPlugHS100 | 1352 |
| D-LinkDayCam | 1235 |
| TP-LinkPlugHS110 | 1229 |
| MAXGateway | 1175 |
| HomeMaticPlug | 1081 |
| Aria | 962 |
| EdimaxCam1 | 525 |
| SmarterCoffee | 242 |
| EdnetCam1 | 241 |
| iKettle2 | 228 |

**Fig. 4.1**: Aalto Dataset

### 4.1.1 Aalto Dataset

The IOT Sentinel dataset (also known as the aalto dataset) is made up of network traffic data from 31 smart home IOT devices. Dataset has devices belonging to 27 types because

4 device types are represented by 2 devices each. Each device setup was done at least 20 times per device type to ensure accurate and robust results. In our analysis, we have used only 25 devices, as shown in the figure 4.1. This selection was made due to the limited amount of available data for the remaining devices.

### 4.1.2 UNSW Dataset

The second dataset we utilized is the UNSW dataset, which consists of network data of 28 IOT devices over a span of 26 weeks. But in our analysis we have used only a part of this data which consists of 17 IOT devices as shown in figure 4.2. Despite this limited selection, it is worth noting that the UNSW dataset is nearly five times greater in size than the Aalto dataset, providing us with a considerable quantity of data from which we can derive insights.

| | |
|---|---|
| Amazon Echo | 313681 |
| Dropcam | 165085 |
| Samsung SmartCam | 67254 |
| Belkin wemo motion sensor | 46585 |
| Withings Smart Baby Monitor | 45825 |
| Belkin Wemo switch | 41510 |
| Smart Things | 29274 |
| Netatmo Welcome | 23184 |
| Netatmo weather station | 13554 |
| TP-Link Day Night Cloud camera | 11625 |
| Triby Speaker | 9288 |
| HP Printer | 8872 |
| Samsung Galaxy Tab | 7418 |
| PIX-STAR Photo-frame | 6658 |
| TP-Link Smart plug | 2962 |
| NEST Protect smoke alarm | 289 |
| Withings Smart scale | 161 |

**Fig. 4.2**: UNSW Dataset

## 4.2 Static Fingerprinting Results

We used five different classifiers, namely Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Naive Bayes, and Decision Tree (DT), for the identification of IoT devices. To ensure robust results, we conducted a evaluation using a 5-fold cross-validation with 10 repetitions.

First we evaluated all models with IOT sentinel features only. The evaluation was performed on two datasets: the Aalto dataset and the UNSW dataset. The table 4.1 presents the evaluation metrics, including Accuracy, Precision, Recall, and F1-score, for each classifier using the IoT Sentinel features on the Aalto dataset.

| | Classifier | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| 0 | NB | 0.074442 | 0.078158 | 0.118028 | 0.041053 |
| 1 | DT_r | 0.697851 | 0.656936 | 0.561841 | 0.591633 |
| 2 | RF | 0.698221 | 0.655179 | 0.562315 | 0.592717 |
| 3 | KNN_R | 0.683068 | 0.643055 | 0.559800 | 0.572166 |
| 4 | SVM | 0.603981 | 0.570921 | 0.503003 | 0.552342 |
| 5 | GB | 0.381497 | 0.053038 | 0.072050 | 0.054111 |

**Table 4.1**: Static Fingerprinting Results
Dataset : Aalto
Features : IOT Sentinel
Classes : All classes

Similarly, the table 4.2 provides the evaluation metrics for each classifier using the IoT Sentinel features on the UNSW dataset.

| | Classifier | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| 0 | NB | 0.468994 | 0.163095 | 0.191749 | 0.155034 |
| 1 | DT_r | 0.905032 | 0.899192 | 0.783157 | 0.820046 |
| 2 | RF | 0.905072 | 0.898631 | 0.783615 | 0.820122 |

**Table 4.2**: Static Fingerprinting Results
Dataset : UNSW
Features : IOT Sentinel

We can see that accuracy of Random forest(RF) for UNSW dataset is much higher than that on aalto dataset, the reason behind less accuracy on aalto dataset maybe the

presence of similar type of devices in aalto dataset. We can verify this hypothesis from the confusion matrix of RF on aalto dataset. We can see that the classifier is misclassifying the classes wich are similar to each other, as indicated by the red marked region in fig 4.3. In the Aalto dataset, some devices have similar purposes and are manufactured by the same company. Additionally, some devices are different models of the same device type. Due to this classifier is not able to distinguish between them and hence the accuracy is low.

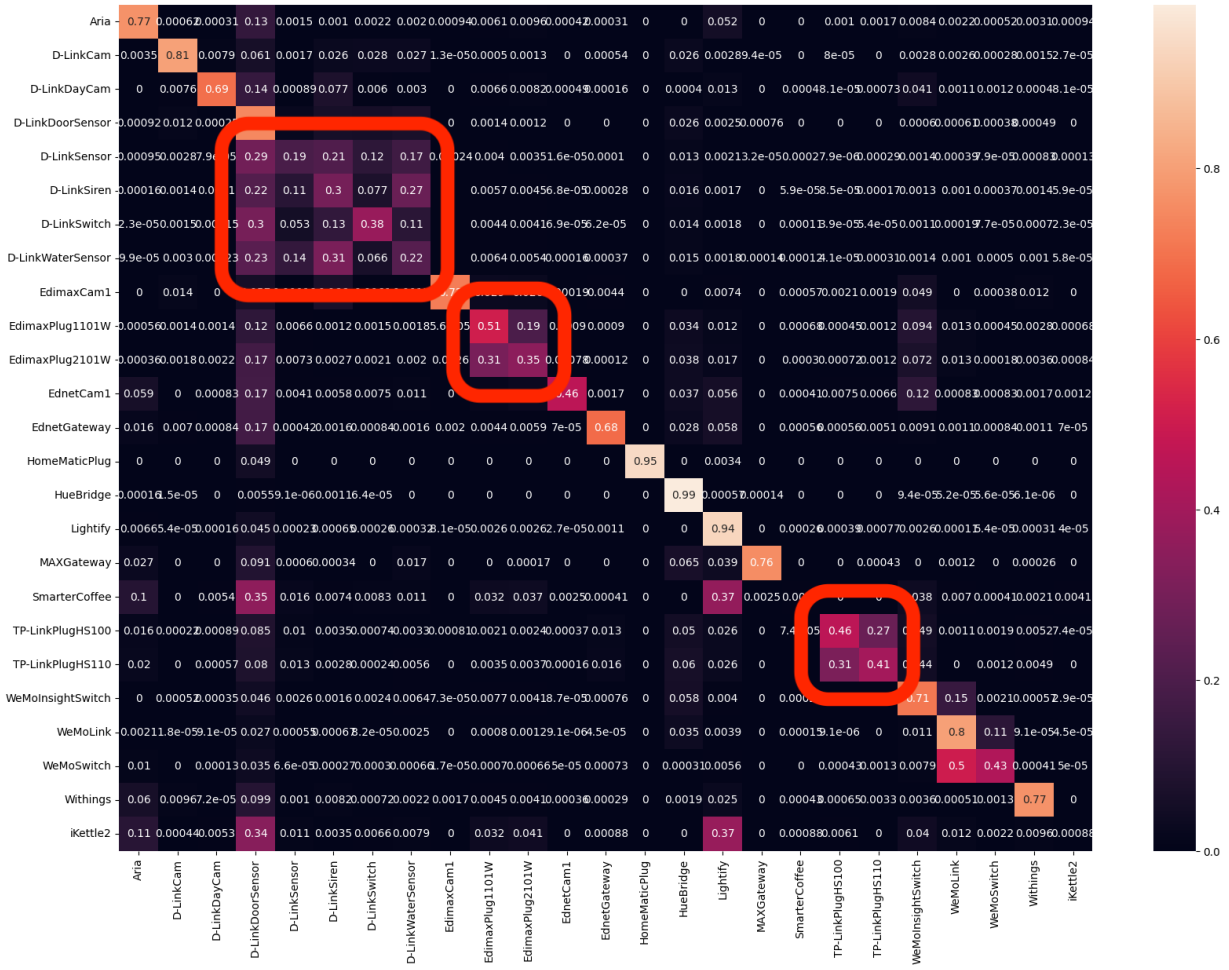



**Fig. 4.3**: Confusion matrix of RF with IOT Sentinel Features

As our main motive is to distinguish between different IOT devices and not between different models of same IOT device, it is reasonable to consider grouping similar devices into the same category. By labeling these similar devices under the same category, the

classification accuracy on the Aalto dataset has increased to 80% as shown in table 4.3

| | Classifier | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| 0 | NB | 0.079214 | 0.101221 | 0.137040 | 0.051249 |
| 1 | DT_r | 0.805266 | 0.834064 | 0.682186 | 0.735433 |
| 2 | RF | 0.804250 | 0.830102 | 0.682095 | 0.734721 |
| 3 | KNN_R | 0.764658 | 0.768569 | 0.677958 | 0.702927 |
| 4 | GB | 0.437592 | 0.067404 | 0.083284 | 0.064692 |

**Table 4.3**: Static Fingerprinting Results
Dataset : Aalto
Features : IOT Sentinel
Classes : Removed Similar classes

By incorporating additional features proposed: TTL, Direction, TCP Flags, Byte Distribution of Payload, TLS Cipher Suites, and TLS Extensions, the accuracy of the RF classifier significantly improved for both the Aalto and UNSW datasets.

| | Classifier | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| 0 | NB | 0.102265 | 0.151979 | 0.228913 | 0.077948 |
| 1 | DT_r | 0.922274 | 0.885527 | 0.882013 | 0.883298 |
| 2 | RF | 0.940567 | 0.936513 | 0.911327 | 0.922707 |

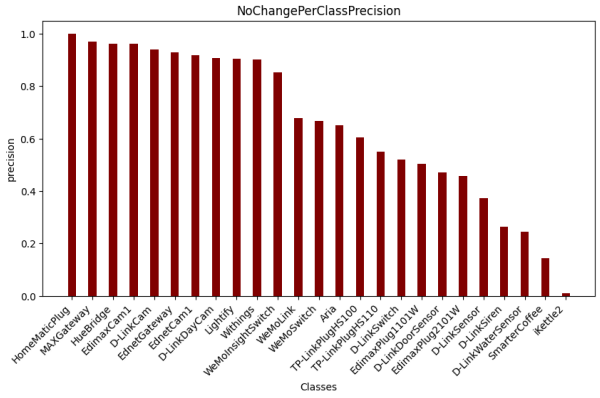
**Table 4.4**: Static Fingerprinting Results
Dataset : Aalto
Features : IOT Sentinel + Added features
Classes : All classes

| | Classifier | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| 0 | NB | 0.103612 | 0.194334 | 0.260430 | 0.099331 |
| 1 | DT_r | 0.943560 | 0.923172 | 0.921021 | 0.921880 |
| 2 | RF | 0.950490 | 0.961667 | 0.933415 | 0.946199 |

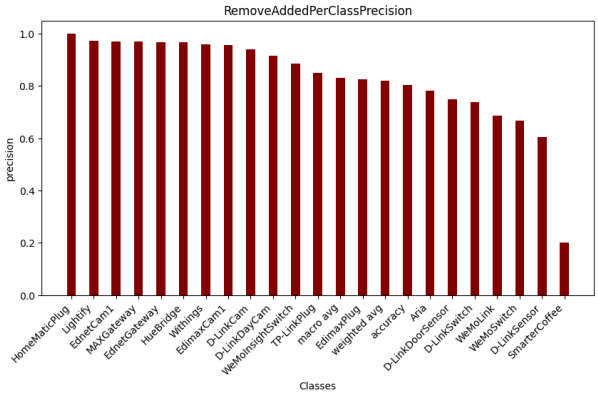
**Table 4.5**: Static Fingerprinting Results
Dataset : Aalto
Features : IOT Sentinel + Added Fetures
Classes : Removed Similar classes

In the case of the Aalto dataset, the RF classifier achieved an accuracy of 94% when considering all the classes. Furthermore, after removing the same classes from the aalto dataset and incorporating the new features, the accuracy further increased by 1% to reach

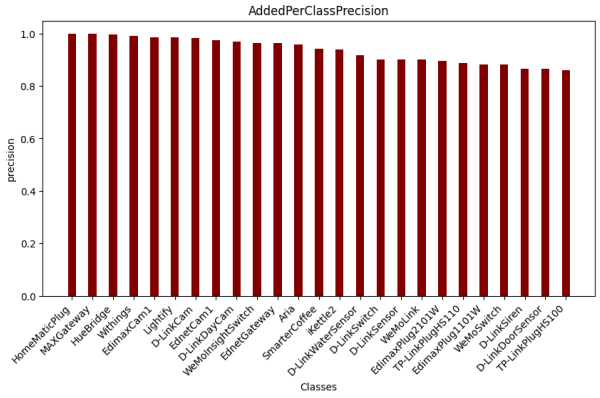
95% (Table 4.5). The increase in accuracy is only 1% with new features, whereas increase was 10% with only IOT sentinel features. This indicates that the new features were successful in distinguishing between similar types of devices and different models of the same device.



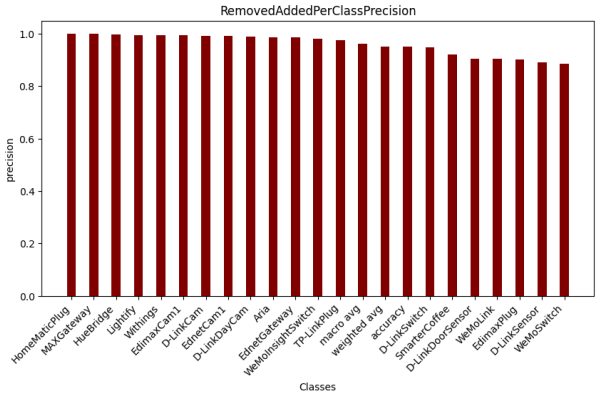

(a) IOT Sentinel Features , All Classes




(b) IOT Sentinel Features , Removed Similar Classes




(c) IOT Sentinel Features + Added Features , All Classes



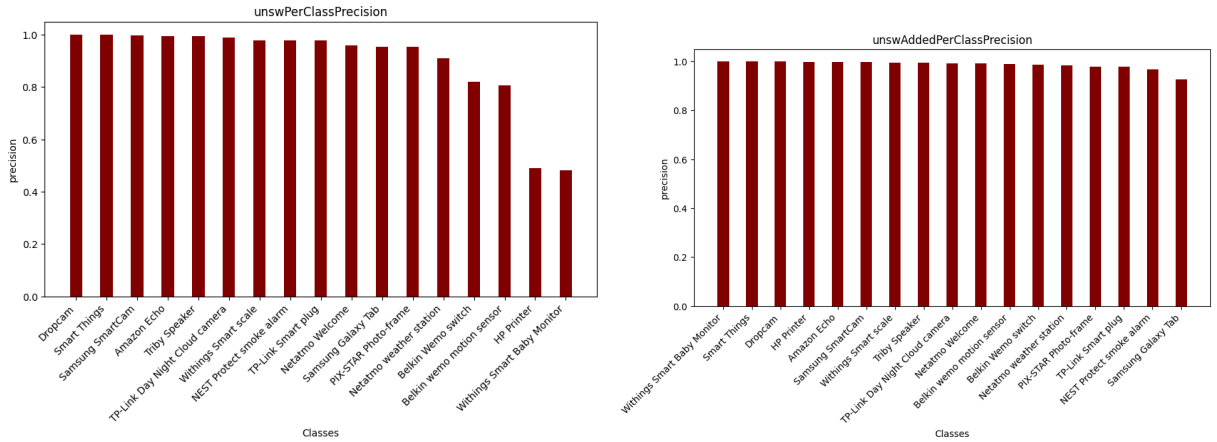

(d) IOT Sentinel Features + Added Features , Removed Similar Classes

**Fig. 4.4**: RF per class Precision plots (Aalto Dataset)

On the other hand, the RF classifier performed exceptionally well on the UNSW dataset, achieving an accuracy of 99.5%. This high accuracy on the UNSW dataset indicates that the new features were highly effective in distinguishing between the different IoT devices present in the dataset.

| | Classifier | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| 0 | NB | 0.406467 | 0.212850 | 0.199984 | 0.165862 |
| 1 | DT_r | 0.993292 | 0.957583 | 0.947802 | 0.951968 |
| 2 | RF | 0.995696 | 0.986735 | 0.975005 | 0.980583 |

**Table 4.6**: Static Fingerprinting Results
Dataset : UNSW
Features : IOT Sentinel + Added Features




(a) IOT Sentinel Features

(b) IOT Sentinel Features + new features

**Fig. 4.5**: RF per class Precision plots (UNSW Dataset)

## 4.3 Dynamic Fingerprinting Results

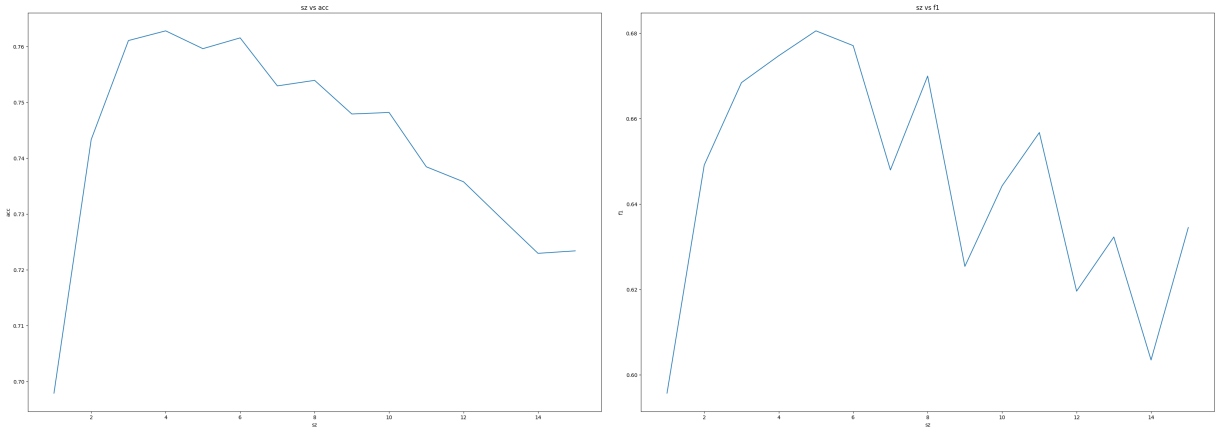
As Random Forest is performing better than all other classifiers on both the datasets in static analysis, we used RF only for further analysis. In dynamic analysis we aggregate the multiple packets to form a single fingerprint.

In order to examine how packet aggregation affects classification accuracy, we trained multiple Random Forest (RF) models with different sizes ranging from 1 to 15 packets, using a 5-fold cross-validation with 10 repetitions to ensure robustness of the results.

Similar to the static analysis, we initially conducted an evaluation using the IoT sentinel features only on both Aalto dataset and the UNSW dataset. To illustrate the relationship between the aggregation size, accuracy and F1-score achieved by the RF model, we have plotted fig 4.6 and fig 4.8.

fig 4.6 presents the plot of size versus accuracy and size vs f1-score for the RF classifier

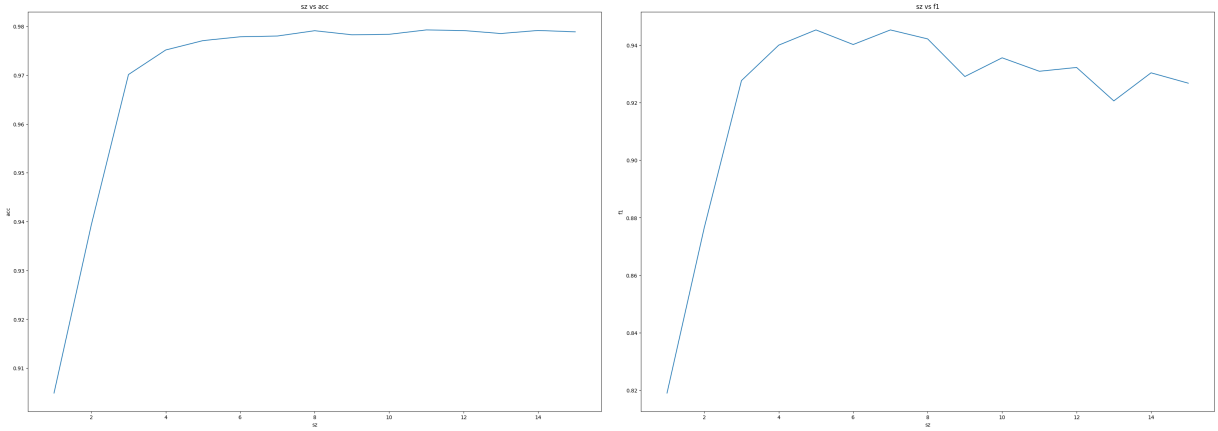
on the Aalto dataset. This graph allows us to visualize how the accuracy, f1-score of the RF model changes as the aggregation size increases.




(a) size vs accuracy (b) size vs f1-score

**Fig. 4.6**: Plot of acc, f1-score with size (aalto Dataset)

Similarly, fig 4.8 depicts the plot of size versus accuracy and size vs f1-score for the RF classifier on the UNSW dataset.



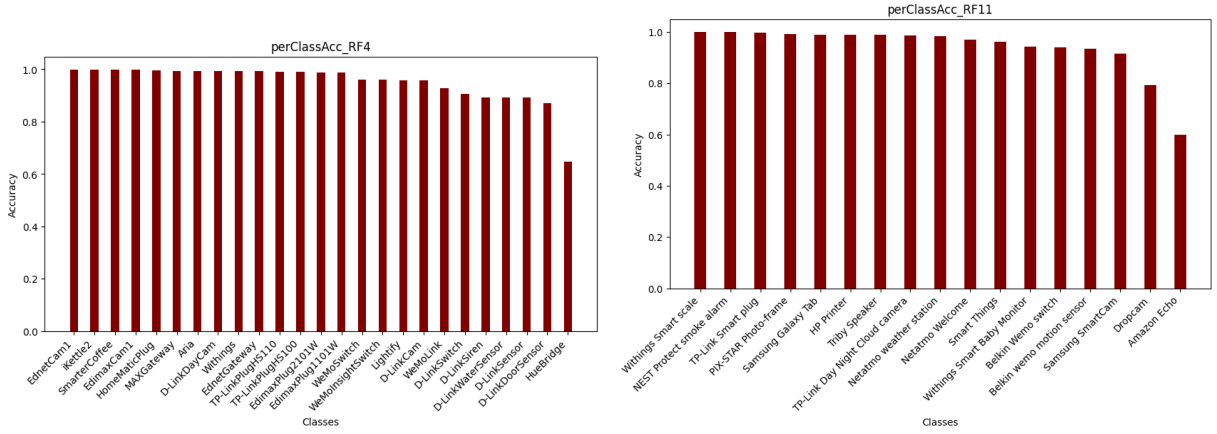

(a) size vs accuracy (b) size vs f1-score

**Fig. 4.7**: Plot of acc, f1-score with size (UNSW Dataset)

We can observe from above graphs a different pattern in the relationship between aggregation size and accuracy on the Aalto dataset, and on UNSW dataset. On aalto dataset the accuracy first increases then decreases. Whereas on the UNSW dataset, the accuracy consistently improves as the aggregation size increases.

One possible explanation for this disparity is the difference in dataset sizes. The Aalto dataset is approximately one-fifth the size of the UNSW dataset. As the aggregation size increases, the number of available training examples decreases proportionally. This reduction in training instances in the Aalto dataset may lead to a decline in accuracy after an initial improvement, as the RF model gets a reduced amount of data to learn from.

Whereas, the larger size of the UNSW dataset allows for an increase in accuracy as the aggregation size grows. As there still will be enough training examples for RF to learn and distinguish between the devices.




(a) aalto (b) UNSW

**Fig. 4.8**: Plot of Per Class Acc for size at which max acc is achieved(IOT Sentinel Features)

Similar to static analysis, RF was not able to distinguis between similar classes which we can verify from confusion matrix shown in fig 4.9.

So, we removed the similar classes by labelling them under same category and trained RF again. And the results after removing similar classes are shown in fig 4.10.

Again we can observe that by incorporating additional features proposed by us: TTL, Direction, TCP Flags, Byte Distribution of Payload, TLS Cipher Suites, and TLS Extensions, the accuracy of the RF classifier significantly improved for both the Aalto and UNSW datasets. Results of the same can be seen in table 4.7

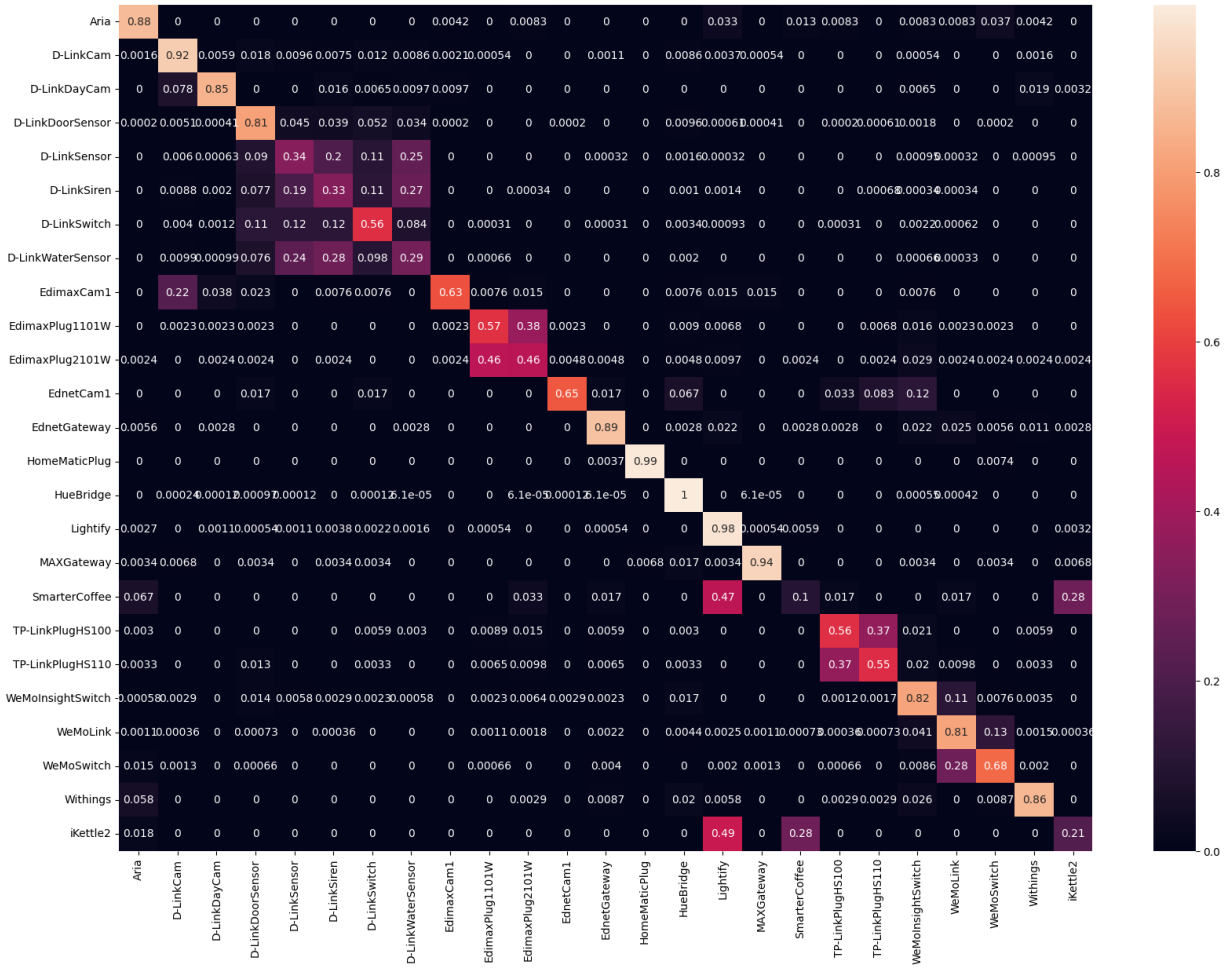
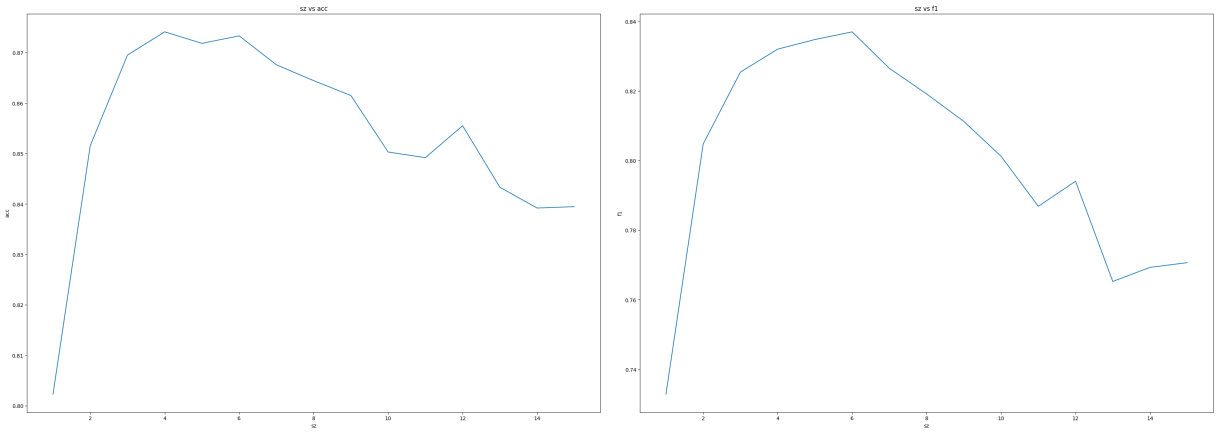



**Fig. 4.9**: Confusion matrix of Dynamic RF with IOT Sentinel Features

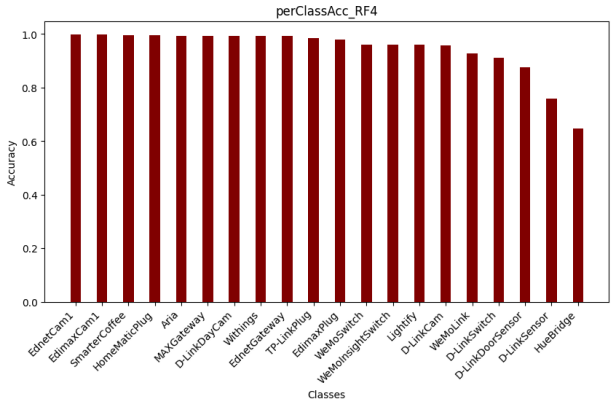
| | | Aalto | UNSW |
|---|---|---|---|
| **All Classes** | **Initial Features** | 0.762809 | 0.979266 |
| | **Added Features** | 0.952819 | 0.996972 |
| **Removed Similar Classes** | **Initial Features** | 0.874135 | — |
| | **Added Features** | 0.964936 | — |

**Table 4.7**: Accuracy of RF after new features are added




(a) size vs accuracy

(b) size vs f1-score




(c) Per Class Acc for sz = 4 (max acc)

**Fig. 4.10**: Plot of acc, f1-score with size after removing similar classes(aalto Dataset)

## 4.4 Static vs Dynamic

In this subsection, we will perform a comparison between the results obtained from static and dynamic fingerprinting methods. Firstly, we will examine the accuracy achieved by both methods on the Aalto and UNSW datasets.

The accuracy on the Aalto dataset increased from 70% to 80% when using dynamic fingerprints, while for the UNSW dataset, the accuracy rose from 90% to 97% when training with the IOT sentinel features, as presented in Table 4.8.

Similarly When classifier is trained using new features the accuracy on the Aalto dataset increased from 94% to 95% when using dynamic fingerprints, while for the UNSW dataset, the accuracy rose from 99.5% to 99.6% as presented in Table 4.8.

| | | Static | Dynamic |
|---|---|---|---|
| **Initial Features** | **Aalto** | 0.698221 | 0.762809 |
| | **UNSW** | 0.905072 | 0.979266 |
| **Initial + New Features** | **Aalto** | 0.940567 | 0.952819 |
| | **UNSW** | 0.995696 | 0.996972 |

**Table 4.8**: Comparing Accuracy of RF using Static and Dynamic method

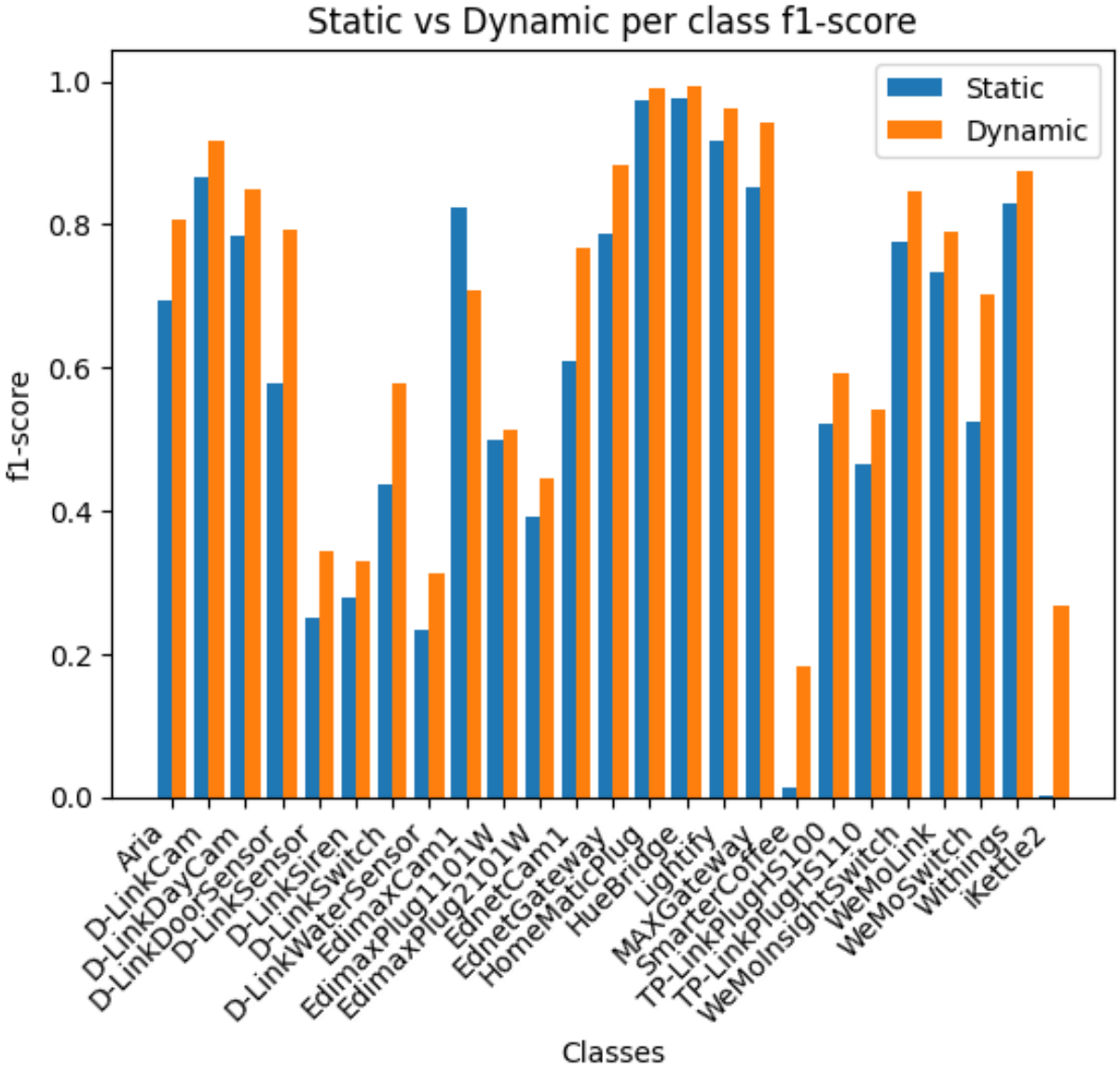



**Fig. 4.11**: Static vs Dynamic per class F1-Score (aalto)

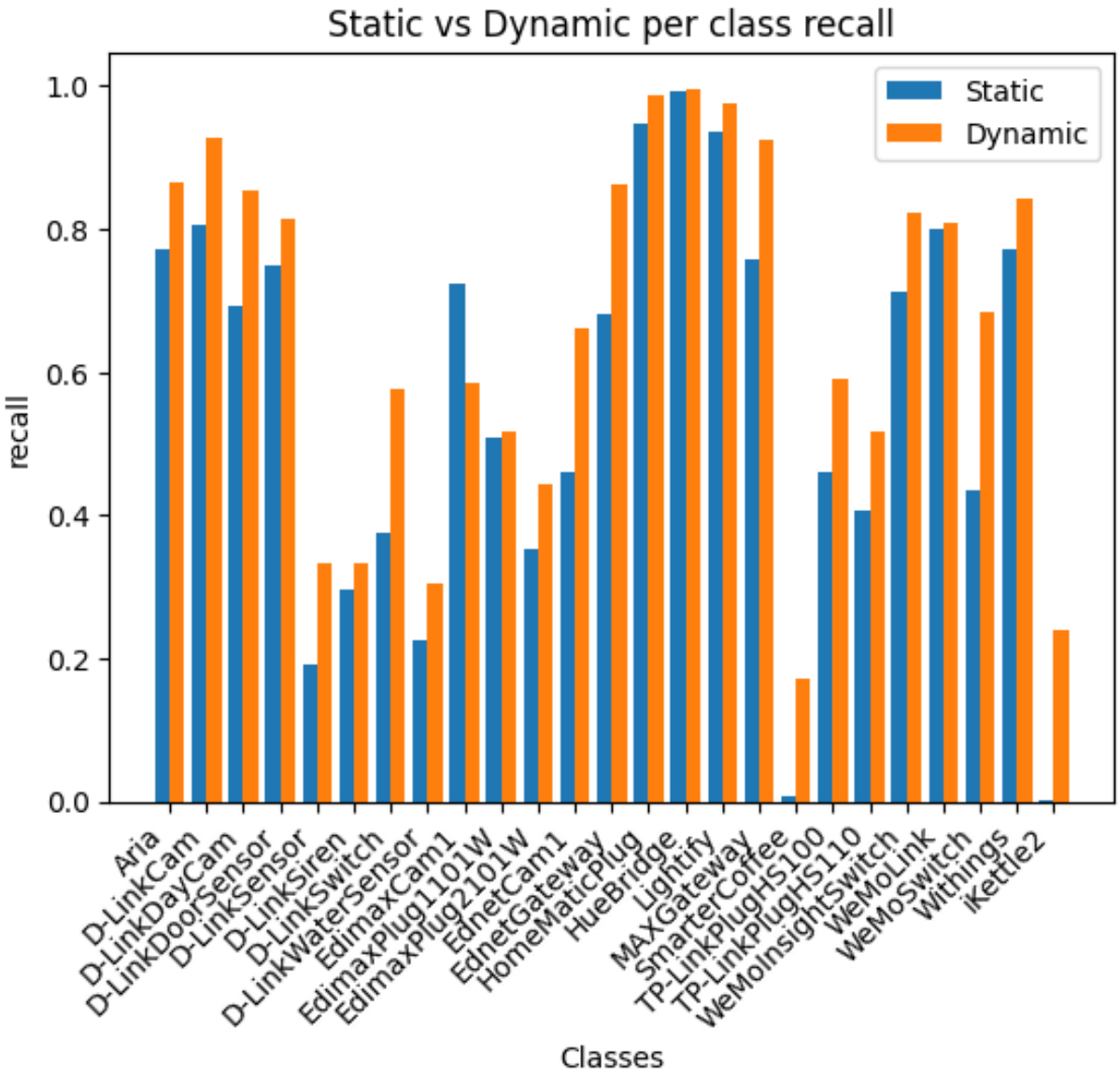



**Fig. 4.12**: Static vs Dynamic per class recall (aalto)

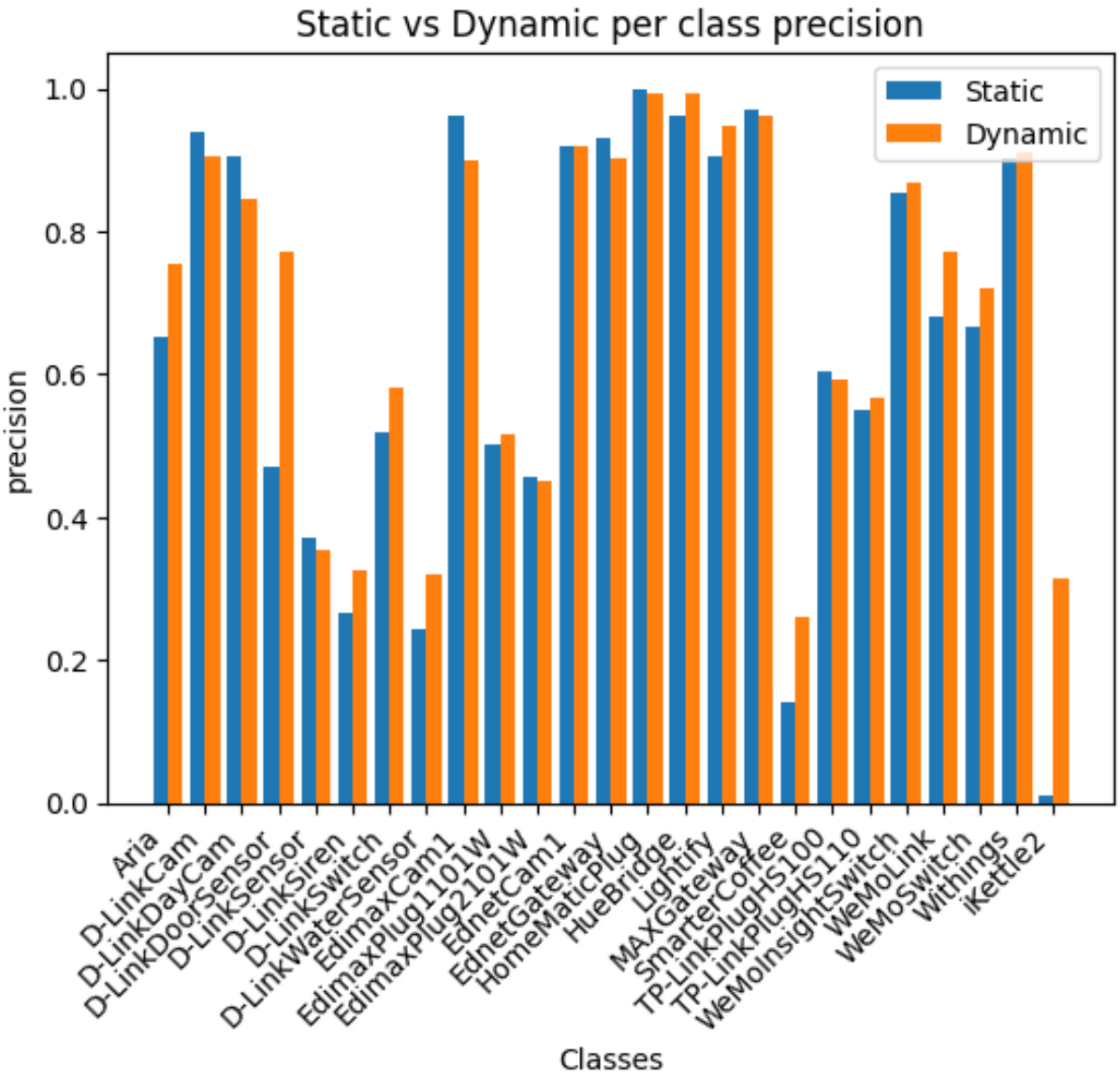



**Fig. 4.13**: Static vs Dynamic per class precision (aalto)

## 4.5 Comparision on different set of Features

In this subsection, we will perform a comparison between the results obtained by training the model on different feature sets.

| Features | Static | Dynamic |
|---|---|---|
| Initial | 0.905072 | 0.979266 |
| TLS | 0.396058 | 0.39793 |
| BD | 0.989583 | 0.991546 |
| Initial + BD | 0.995696 | 0.996972 |
| Initial + BD + TLS | 0.995313 | 0.996306 |

**Table 4.9**: Comparision of Accuracy of Random Forest on different sets of feature

Table 4.9 compares the Accuracy of Random Forest on different sets of feature. Similar observation can be made through fig 4.14 which shows that many byte distribution features are in top 20 features deemed important by the Random Forest algorithm. This, implies that byte distribution plays an important role in distinguishing and identifying IOT devices.

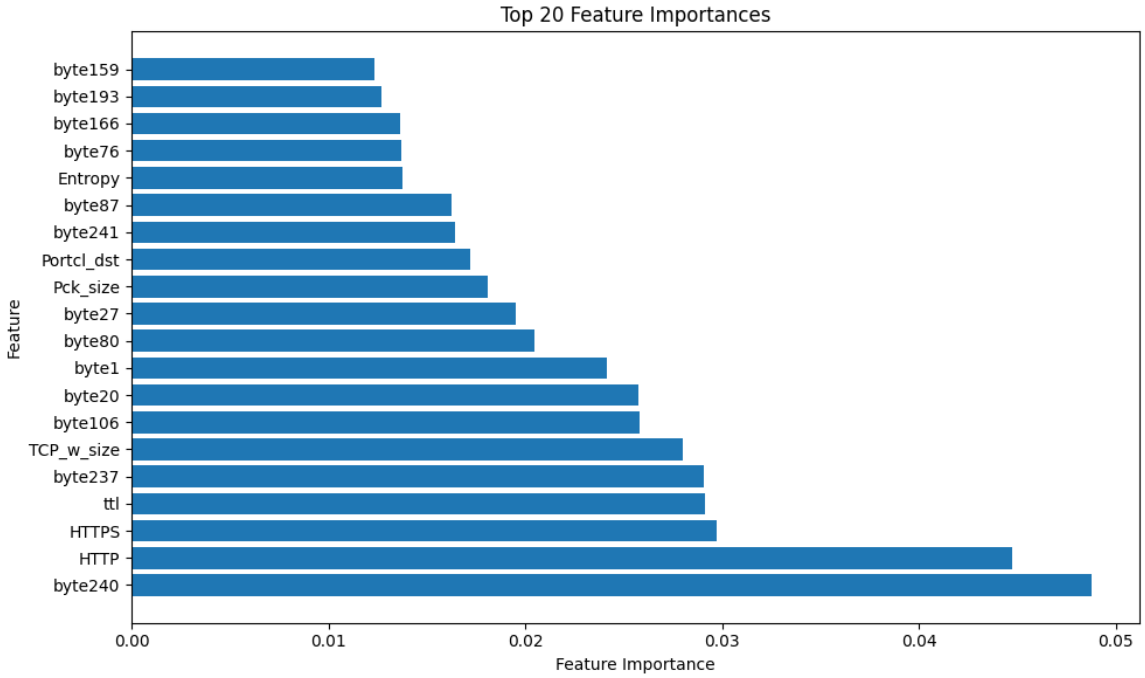



**Fig. 4.14**: Feature Importance

# Chapter 5

# Conclusion and Future Work

While our analysis has provided valuable insights into the fingerprinting and identification of IoT devices using network traffic data, there are several areas that offer potential for future research and development. These include:

1. Adding New Features :

Exploring additional features or improving existing feature extraction techniques can further enhance the accuracy and effectiveness of device fingerprinting.

2. Real-Time Fingerprinting:

The development of real-time fingerprinting algorithms can allow for continuous monitoring and identification of IoT devices while they function in a network. This can include monitoring and analysing real-time network traffic, enabling for quick detection and reaction to new or unknown IoT devices.

3. Deep Learning Techniques:

Deep Learning has shown a great potential in various Fields. So, building more DL architectures, such as CNNs or RNNs, can further improve the accuracy and performance of IoT device identification.

In the static analysis, the evaluation was done using many classifiers using IoT Sentinel features, and RF consistently performed the best on both the Aalto and UNSW datasets.

However, it was observed that the accuracy on the Aalto dataset was comparatively lower due to the presence of similar classes. By labeling similar classes under the same category, the accuracy on the Aalto dataset was significantly improved.

Dynamic analysis involves aggregating multiple packets to form a single fingerprint. RF models were trained with different aggregation sizes. The results showed that using dynamic fingerprinting methods, led to an increase in accuracy for both datasets. The accuracy on the Aalto dataset improved from 70% to 80% using dynamic fingerprints, while the accuracy on the UNSW dataset using IoT Sentinel features increased from 90% to 97%.

The incorporation of additional features, including TTL, Direction, TCP Flags, Byte Distribution of Payload, TLS Cipher Suites, and TLS Extensions, further improved the accuracy of the RF classifier for both datasets. For aalto dataset we achieved and accuracy of 96% and for UNSW dataset we achieved 99% accuracy. Therefore it can be concluded that Random Forest with addedfeatures and dynamic fingerprint method gives best accuracy.

In conclusion, the analysis focused on the identification of IoT devices using a static and dynamic fingerprinting methods. The Random Forest (RF) classifier was found as the most effective model for this task, outperforming other classifiers such as SVM, KNN, Naive Bayes, and Decision Trees.

# References


[1] IoT Analytics. State of iot 2021: Number of connected iot devices growing 9% to 12.3 billion globally, cellular iot now surpassing 2 billion. *URL: https://iot-analytics.com/number-connected-iot-devices*, 2021.

[2] Sandhya Aneja, Nagender Aneja, and Md Shohidul Islam. Iot device fingerprint using deep learning. In *2018 IEEE international conference on internet of things and intelligence system (IOTAIS)*, pages 174–179. IEEE, 2018.

[3] Lei Bai, Lina Yao, Salil S Kanhere, Xianzhi Wang, and Zheng Yang. Automatic device classification from network traffic streams of internet of things. In *2018 IEEE 43rd conference on local computer networks (LCN)*, pages 1–9. IEEE, 2018.

[4] Bruhadeshwar Bezawada, Maalvika Bachani, Jordan Peterson, Hossein Shirazi, Indrakshi Ray, and Indrajit Ray. Behavioral fingerprinting of iot devices. In *Proceedings of the 2018 workshop on attacks and solutions in hardware security*, pages 41–50, 2018.

[5] Kahraman Kostas, Mike Just, and Michael A Lones. Iotdevid: A behaviour-based fingerprinting method for device identification in the iot. *arXiv preprint arXiv:2102.08866*, 2021.

[6] Kahraman Kostas, Mike Just, and Michael A Lones. Iotdevid: A behavior-based device identification method for the iot. *IEEE Internet of Things Journal*, 9(23):23741–23749, 2022.

[7] Yair Meidan, Michael Bohadana, Asaf Shabtai, Martin Ochoa, Nils Ole Tippenhauer, Juan Davis Guarnizo, and Yuval Elovici. Detection of unauthorized iot devices using machine learning techniques. *arXiv preprint arXiv:1709.04647*, 2017.

[8] Markus Miettinen, Samuel Marchal, Ibbad Hafeez, N Asokan, Ahmad-Reza Sadeghi, and Sasu Tarkoma. Iot sentinel: Automated device-type identification for security enforcement in iot. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 2177–2184. IEEE, 2017.

[9] Nizar Msadek, Ridha Soua, and Thomas Engel. Iot device fingerprinting: Machine learning based encrypted traffic analysis. In *2019 IEEE wireless communications and networking conference (WCNC)*, pages 1–8. IEEE, 2019.

[10] Chao Shen, Ruiyuan Lu, Saeid Samizade, and Liang He. Passive fingerprinting for wireless devices: A multi-level decision approach. In *2017 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*, pages 1–6. IEEE, 2017.

[11] Sandra Siby, Rajib Ranjan Maiti, and Nils Tippenhauer. Iotscanner: Detecting and classifying privacy threats in iot neighborhoods. *arXiv preprint arXiv:1701.05007*, 2017.

[12] Arunan Sivanathan, Hassan Habibi Gharakheili, Franco Loi, Adam Radford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman. Classifying iot devices in smart environments using network traffic characteristics. *IEEE Transactions on Mobile Computing*, 18(8):1745–1759, 2018.