# Traffic Classification of Home Network Devices using Supervised Learning

Adriano A. M. De Resende, Pedro H. A. D. De Melo, Jefferson R. Souza, Renan G. Cattelan and Rodrigo S. Miani

*Faculty of Computing, Federal University of Uberlândia, Uberlândia, Brazil*

Keywords: Computer Networks, Traffic Classification, Internet of Things, Machine Learning, Supervised Learning.

Abstract: Network traffic classification is a relevant tool for computer network management. In the last decade, researchers have been adopting machine learning algorithms to identify different types of traffic in a network. Traffic classification can be used to identify threats and improve the quality of service of networks. Literature in this area usually focuses on using network flows to identify the traffic of specific devices, for example, IoT devices. This paper proposes a network traffic classification model to identify IoT smart home devices and personal computers (PCs). The idea is to evaluate the performance of decision models trained with different devices to identify IoT and non-IoT network traffic. We created two scenarios to mimic the behavior of a home network. In the first scenario, we evaluate how training a model with only PC devices influences the identification of IoT and non-IoT traffic. The second one attempts to assess how well the network traffic of a brand new type of IoT device could be identified using supervised learning. Our results show that the supervised models were able to identify the network traffic; however, their performance varies across the algorithms.



## 1 INTRODUCTION

Traffic classification is a problem that can benefit two aspects of computer networks, security and the quality of service (QoS) (Cherif and Kortebi, 2019). Two of the classical approaches to this problem are port-based classification and payload inspection. The first one has the shortcoming of not being able to perform well in respect of some applications that can run in any port number (Karagiannis et al., 2004). The payload-based technique, on the other hand, is not able to deal with encrypted traffic (Al Khater and Overill, 2016). One alternative to mitigate the previous limitations is the use of flow traces. A flow can be defined as a set of IP packets passing an observation point in the network during a specific time interval (Sperotto et al., 2010). All packets belonging to a particular flow have a set of standard features. Usually, every feature is defined by applying a function to the values of one or more packet header fields. The main idea of using flow traces is to use such features to classify the network traffic. The use of machine learning algorithms for classifying network flows became widespread in the last years (Soysal and Schmidt, 2010; Holbrook and Alamaniotis, 2019).

One of the potential implications of the network traffic classification field would be monitoring traffic on a home network to protect it. A home network encompasses devices in a home that connect to a router (gateway), which in turn connects to an Internet provider responsible for ensuring the router has Internet access (Kurose and Ross, 2016). Two types of devices can be associated with a home network: i) personal computers such as laptops and smartphones and ii) IoT intelligent home devices, which can be defined as any single-purpose Internet-connected device intended for home use (Apthorpe et al., 2017). Some examples of smart home devices include thermostats, bulbs, and smart speakers.

Considering that the home computing environment could be divided into IoT devices (smart-TVs, smart-speakers, and cameras) and non-IoT devices (mobile devices and personal computers (PCs)), we would like to know whether such traffic could be identified, and also the impact of each traffic type on the decision model. First, we created a dataset consisting of the network traffic of these three types of devices. Second, we divided this dataset into multiple smaller datasets to represent two scenarios. Lastly, we applied different machine learning algorithms in each of these cases and analyzed their results. Therefore, our goal is to investigate the use of supervised machine learn-



M. De Resende, A., D. De Melo, P., Souza, J., Cattelan, R. and Miani, R.
Traffic Classification of Home Network Devices using Supervised Learning.
DOI: 10.5220/0010785500003116
In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 3*, pages 114-120
ISBN: 978-989-758-547-0; ISSN: 2184-433X

ing techniques to identify IoT and non-IoT traffic in the context of home networks.

Our contributions are twofold. First, we introduce a labeled dataset with home network traffic for machine learning tasks. Second, we evaluated the performance of several supervised machine learning algorithms on two home network classification problems: i) learning the impact of different devices (mobile and PCs) on separating IoT from non-IoT traffic and ii) identification of a new IoT device type. Some applications of our approach include developing specialized anomaly detection systems and also helping ISP providers understand the network traffic behavior of home networks.

The remainder of this paper is organized as follows. Section 2 focuses on related work in the area of network traffic classification; Section 3 introduces our approach to the problem, detailing how data was collected, the data processing steps, and the machine learning algorithms used; Section 4 describes our results; and, finally, Section 5 presents final remarks and future work.

## 2 RELATED WORK

Overall, research focusing on network traffic classification tries to identify the application or the device that originated the traffic. Next, we review some of these studies, briefly overview their objectives, and present the main classification algorithms used and types of device traffic considered.

In (Lashkari et al., 2017), the authors used traffic classification techniques to identify applications encrypted by the Tor browser. They hypothesize that time-based features are enough to determine its application of origin. They focus mainly on the home network traffic of desktop applications. They used K-nearest neighbors, Random Forests, and C4.5 algorithms for the classification process.

In (Cherif and Kortebi, 2019), the authors aim to verify the performance of the eXtreme Gradient Boosting algorithm on home network traffic classification. They use a real-world dataset obtained from a French ISP, but there is no reference to the type of devices (IoT, mobile, or PCs). Their idea is to identify the application that originated the traffic. Since their classification traffic is derived from the following applications: BitTorrent, Facebook, Google, HTTP, HTTPS, Quic, and Skype as classes, we can assume the existence of PC or mobile devices traffic. They compare the XGBoost result with K-nearest neighbors, Naive Bayes, C4.5, and C5.0.

A Tree-structured recurrent neural network for traffic classification is proposed in (Ren et al., 2021). A key feature of their approach relies on the fact that their method can work directly with encrypted traffic. The dataset used for their model evaluation is a public dataset created by (Draper-Gil et al., 2016). That dataset consists of PC applications' traffic.

All the above studies focus on identifying traffic from specific applications. However, there exist some efforts from identifying IoT devices by identifying their network traffic. (Tahaei et al., 2020) presents a survey of traffic classification solutions for different IoT network problems. The issues raised by the authors are anomaly detection, IoT devices identification, IoT devices behavior monitoring, traffic monitoring, user authentication model, low-power massive M2M communications, smart-city traffic classification, and IoT-based intelligent health care systems.

(Shahid et al., 2018) is one of the studies focused on IoT device identification. The authors used a dataset to represent an intelligent home containing four devices: a Nest security camera, a D-Link Motion Sensor, an TP-Link Smart Bulb, and a TP-Link Smart Plug. They used the following classification algorithms Random Forests, Decision Tree, Support Vector Machine, K-nearest neighbors, Artificial Neural Network, and Gaussian Naive Bayes.

In (Cvitić et al., 2021) the authors used IoT device Traffic to classify smart home IoT devices (SHIoT) in one of four defined classes. They set up an environment with 41 SHIoT devices. Some of these devices included smart bulbs, smart assistants, cameras, and others. The traffic used as data for the classification algorithm was the ones captured in this environment. The algorithm they chose to use was the logistic regression method enhanced by the concept of supervised machine learning (logitboost).

(Liu et al., 2021) reviews the literature that focuses on the use of machine learning for IoT device identification and compromised device identification. The authors present four categories of problems, i) device-specific pattern recognition, ii) Deep Learning enabled device identification, iii) unsupervised device identification and iv) abnormal device detection. In pattern recognition were presented works that made use of features to represent the IoT traffic. Some of the algorithms used in these problems were Naive Bayes and Random Forests. Deep Learning enabled device identification is associated with studies that use models capable of working with raw data instead of generating features to represent the traffic. Some of the algorithms included Convolutional Neural Network, Convolutional Deep Complex-valued Neural Network, and Recurrent Deep Complex-valued Neural Network. Unsupervised device identification

present studies that used unsupervised techniques for IoT device identification such as one-class clustering and Long Short Term Memory enabled autoencoder. Lastly, the abnormal device detection category focuses on works aiming to detect devices with abnormal behavior; Markov models, deep autoencoders, and Convolutional Neural Networks were some of the algorithms used in different papers of this area.

As mentioned before, most of the related work in the area can identify either the application that originated the traffic or the IoT device. No other study presents traffic classification for different categories of devices in the innovative home context as far as we are concerned.

# 3 METHODS

In this section, we present our dataset, as well as the machine learning algorithms used. In synthesis, the first step consisted of obtaining data to represent the traffic for the classes of target devices: IoT intelligent devices and non-IoT devices. The second step was to convert the collected data to a format more accessible to use, removing undesirable features and splitting the dataset into smaller subsets to represent better the two different. The third step consists of choosing proper classification algorithms and creating the decision models.

## 3.1 Dataset Generation

To create a dataset to represent the traffic of a home network, we merged traffic from three different types of devices: *(i)* IoT smart devices; *(ii)* mobile devices, consisting of smartphones and tablets, running Android or IOS; *(iii)* PCs, including desktop and laptop computers, running Windows, Linux or macOS.

The IoT traffic was generated using the dataset presented in (Sivanathan et al., 2019). Table 1 shows each IoT device and its respective category.

Mobile device data was obtained from (Lashkari et al., 2018). It represents network traffic for mobile applications like GPS, mobile browsing, SMS, and similar ones. Finally, the traffic from PCs was gathered using the dataset proposed in (Ahmed et al., 2016). Here, we have network traffic from different applications, such as P2P applications, audio streaming, browsing, and e-mail.

## 3.2 Dataset Preparation

All the data obtained was in the PCAP format. PCAP is a format used to represent network packets. Since

Table 1: IoT categories and devices (Sivanathan et al., 2019).

| Category | Devices |
|---|---|
| Appliances | HP Envy Printer<br>Google Chromecast<br>Hello Barbie<br>Pixstar Photo Frame<br>Triby Speaker |
| Cameras | Ring Doorbell<br>August Doorbell<br>Netatmo Camera<br>Canary Camera<br>Drop Camera<br>Samsung Smart Cam<br>Withings Baby Monitor<br>TP Link Camera<br>Belkin Camera |
| Controller Hubs | Amazon Echo<br>Hue Bridge<br>Samsung SmartThings |
| Energy Management | Belking Motion Sensor<br>LIFX Lightbulb<br>Phillip Hue Lightbulb<br>iHome Power plug<br>Belkin Switch<br>TP Link Power plug |
| Health Monitor | Withings Sleep Sensor<br>Blipcare BP meter<br>Awair Air Quality Monitor<br>Netatmo weather station<br>Nest Smoke Alarm<br>Withings Scale |

PCAP is a challenging format to work in, a conversion was necessary. We decided to convert the network packets into network flows. Flow-based classifiers are a tendency in the field of network management due to the limitations of tools for analyzing individual network packets. The CICFlowmeter tool(Lashkari et al., 2017) was used in this process. CICFlowmeter identifies traffic flows in a PCAP file and converts them to CSV, calculating more than 80 features for each flow.

The next step was cleaning the converted data. We performed the following actions:

1. Columns with zero variance were removed;
2. The *Flow ID* the column created by CICFlowmeter to identify the flow was removed;
3. Src IP e Dst IP columns were removed. The IP address is a logical address that identifies a host in a network. This information is removed because we don't want our model to learn based on a specific machine;
4. Flows with at least one missing value were re-

moved. Only three flows in the entire dataset generated had columns with at least one missing value. Since this could be related to a conversion problem from CICFlowmeter, we choose to remove them;

5. Flows with wrong data type were removed. Some flows had string values instead of integer values in some attributes.

After the data cleaning, we obtained a database with 1,043,613 IoT flows, 1,140,338 mobile flows, and 828,451 desktop/laptop flows. To train each algorithm, we did not use the entire database. Instead, we randomly selected flows from the database to create our validation datasets. Each dataset has a proportion of device flows according to the two scenarios we created. These scenarios are described as follows:

- Scenario 1 - learning the impact of mobile and PCs on separating IoT from non-IoT traffic. The idea here simulate home networks containing different types of devices: a) IoT smart devices + non-IoT devices (mobile + desktop/laptop), b) IoT smart devices + non-IoT devices (mobile) and c) IoT smart devices + non-IoT devices (desktop/laptop). We generated one dataset for these three cases.
- Scenario 2 - adding a new IoT device. The main idea is to verify what happens to the decision model if a new IoT device is added to the home network. To mimic this situation, we created five datasets. For every dataset, one IoT device category (see Table 1) is suppressed from the training data. The test dataset is composed of samples of the removed IoT category and mixed non-IoT traffic.

For Scenario 1, we have one primary dataset for each type of home network. We call every home network a Training Case (TC). TC 1 represents the network traffic of IoT intelligent devices + non-IoT devices (mobile + desktop/laptop), TC 2 illustrates the network traffic of IoT intelligent devices + non-IoT devices (mobile), and TC 3 represents the network traffic of IoT intelligent devices + non-IoT devices (desktop/laptop). Table 2 presents the number of flows per device in each TC. We created five different training/test datasets by randomly selecting network flows from our original dataset. The experimental results, presented in the next section, represent an average of the five datasets.

For Scenario 2, we have five different datasets: Appliances, Cameras, Controller Hubs, Energy Management, and Health monitors. The Appliances dataset, for example, is divided into training and test. The training set contains Cameras, Controller Hubs, Energy Management, Health Monitor, and non-IoT traffic network flows. In contrast, the test set contains only traffic from Appliances and non-IoT devices (mobile and PC). Both training and test sets have 22,000 network flows, 11,000 IoT smart devices, and 11,000 non-IoT traffic. The organization of the remaining four datasets (Cameras, Controller Hubs, Energy Management, and Health Monitor) follows the same idea.

Table 2: Scenario 1 - Training datasets.

| Training Case | IoT | Mobile | Desktop |
|---|---|---|---|
| TC 1 | 11000 | 5500 | 5500 |
| TC 2 | 11000 | 11000 | 0 |
| TC 3 | 11000 | 0 | 11000 |

### 3.3 Machine Learning Classifiers

We used ten different classification algorithms in this study: Decision Trees (DT), K-nearest neighbors (KNN), Linear Discriminant Analysis (LDA), Multilayer Perceptron (MLP), Quadratic Discriminant Analysis (QDA), Ada Boosting (ADA), Gradient Boosting (GB), Random Forests (RT) and eXtreme Gradient Boosting (XGB). Two factors influenced the composition of the final list: algorithms introduced in related work and algorithms presented in classification competitions. We adopted the Python scikit-learn library for the evaluation tests using only the default values provided by the library. We use the Decision Trees feature selection algorithm and cross-validation with three folds during the training phase. The performance evaluation of each algorithm is based on accuracy and F1-score.

## 4 RESULTS

As mentioned in Section 3 we considered two home network scenarios for classification. This section presents the results obtained in each one of them.

### 4.1 Identifying the Impact of Mobile and PC Traffic on Classifying IoT and non-IoT Traffic

This scenario examines whether there is any change in the classifiers if we train a model using only mobile or PC flows as part of the non-IoT class. As mentioned in Section 3 we have three TCs. All algorithms were trained with all TC datasets and then evaluated on the test datasets.

TC 1 can be viewed as our baseline since it contains all kinds of network traffic. We compare the results obtained in TC 1 with TC 2 e TC 3 to understand what happened when we removed a device from the training set. Since each TC has multiple training datasets, Table 3 presents the mean accuracy and F1-score of all evaluation tests.

Table 3: Experiment 1 - Impact of mobile and PC traffic on classifying IoT and non-IoT traffic.

| Algorithm | Training Case | Accuracy | F1 |
|---|---|---|---|
| ADA | TC 1 | 0.97434 | 0.97440 |
| | TC 2 | 0.90781 | 0.91363 |
| | TC 3 | 0.91820 | 0.92400 |
| DT | TC 1 | 0.99068 | 0.99068 |
| | TC 2 | 0.92464 | 0.92922 |
| | TC 3 | 0.95377 | 0.95572 |
| GB | TC 1 | 0.98622 | 0.98620 |
| | TC 2 | 0.93015 | 0.93402 |
| | TC 3 | 0.93117 | 0.93547 |
| GNB | TC 1 | 0.59437 | 0.47793 |
| | TC 2 | 0.59206 | 0.54045 |
| | TC 3 | 0.56620 | 0.27702 |
| KNN | TC 1 | 0.88870 | 0.88928 |
| | TC 2 | 0.81741 | 0.82983 |
| | TC 3 | 0.79411 | 0.81925 |
| LDA | TC 1 | 0.78382 | 0.79623 |
| | TC 2 | 0.77760 | 0.80363 |
| | TC 3 | 0.74830 | 0.77536 |
| MLP | TC 1 | 0.87201 | 0.87472 |
| | TC 2 | 0.79285 | 0.79729 |
| | TC 3 | 0.78555 | 0.81302 |
| QDA | TC 1 | 0.77891 | 0.74843 |
| | TC 2 | 0.59093 | 0.56013 |
| | TC 3 | 0.76801 | 0.71671 |
| RF | TC 1 | 0.99362 | 0.99362 |
| | TC 2 | 0.93039 | 0.93444 |
| | TC 3 | 0.94088 | 0.94419 |
| XGB | TC 1 | 0.99466 | 0.99466 |
| | TC 2 | 0.92713 | 0.93170 |
| | TC 3 | 0.94366 | 0.94661 |

By inspecting Table 3, it is possible to notice that the mean accuracy and F1-score for the algorithms in TC 2 e TC 3 are lower than the results obtained in TC 1. To verify this difference, we ran two Paired t-Tests; the first compares TC 1 to TC 2, and the second one compares TC 1 and TC 3. For both tests, with a significance of 0.05, we have that the mean of TC 1 is different from TC 2 and TC 3 and the same applies for the F1-score. This result confirms that removing one of the device types of the non-IoT class will impact the classifier result. In other words, considering different types of non-IoT traffic would help increase the classifier's performance. This result also suggests that some non-IoT traffic, when analyzed individually, might be confused with IoT traffic. Therefore, enhancing the non-IoT traffic mix benefits the classifier.

It is important to notice that the only algorithms that had a different behavior were GNB e LDA. In both, they achieved a better F1-score in TC2. By analyzing others metrics, it was possible to see that this increase was related to a better recall. This indicates that not training these algorithms with PC traffic can improve IoT recognition.

## 4.2 Adding a New IoT Device

This scenario verifies what happened when we added an IoT device from a class that wasn't present in the training dataset. As presented in Section 3 we have five different datasets, one for each IoT category. Besides, to compare the impact of removing one type of IoT smart device, we created five classifications models using the following schema. The training dataset contains the five IoT categories and non-IoT traffic, while the test set contains only one IoT category. These models will form our baseline. We call these datasets B_*, where * denotes the name of the IoT category, for example, B_Appliances, B_Cameras, etc.

Table 4 presents the performance of the algorithms trained with our baseline. It is possible to see that some algorithms (ADA, DT, GB, RF, and XGB) achieved excellent results in identifying IoT traffic of only one type using a training model containing all IoT device types. Next, we want to evaluate the network classification performance when removing an IoT device type from the training process. Table 5 presents the accuracy and F1-score for the algorithms when trained with each dataset: Appliances, Cameras, Controller Hubs, Energy Management, and Health Monitor.

When comparing the entrees of Tables 4 and 5, it is possible to see that the performance of the classification model decreases for most IoT device types. The camera device is an example of this behavior. For the KNN algorithm, for instance, we see a decrease of around 16% of accuracy. This result suggests that adding a new IoT device type might be a difficult task for supervised algorithms. We ran two Paired t-Test on each training case to better understand this behavior, one for the accuracy and the other for the F1-Score. We compare the baseline results (Table 4) with the results obtained from the training cases that do not contain flows of the added device (Table 4).

In both tests, we discarded the null hypothesis when comparing Appliances, Cameras, and Controller Hubs. On the other hand, we did not discard the null hypothesis for Energy Management and Health Monitor classes. This result indicates that IoT

Table 4: Experiment 2 - Adding a new smart home IoT device (baseline).

| Algorithm | Training case | F1 | Accuracy |
|---|---|---|---|
| ADA | B_Appliances | 0.99939 | 0.99939 |
| | B_Cameras | 0.99963 | 0.99963 |
| | B_Controller Hu. | 0.99980 | 0.99980 |
| | B_Energy Man. | 0.99982 | 0.99982 |
| | B_Health Mon. | 0.99981 | 0.99981 |
| DT | B_Appliances | 0.99926 | 0.99926 |
| | B_B_Cameras | 0.99971 | 0.99971 |
| | B_Controller Hu. | 0.99982 | 0.99982 |
| | B_Energy Man. | 0.99981 | 0.99981 |
| | B_Health Mon. | 0.99983 | 0.99983 |
| GB | B_Appliances | 0.99934 | 0.99934 |
| | B_Cameras | 0.99966 | 0.99966 |
| | B_Controller Hu. | 0.99983 | 0.99983 |
| | B_Energy Man. | 0.99983 | 0.99983 |
| | B_Health Mon. | 0.99983 | 0.99983 |
| GNB | B_Appliances | 0.82142 | 0.78259 |
| | B_Cameras | 0.77562 | 0.71070 |
| | B_Controller Hu. | 0.84315 | 0.81397 |
| | B_Energy Man. | 0.88022 | 0.86391 |
| | B_Health Mon. | 0.83360 | 0.80037 |
| KNN | B_Appliances | 0.97648 | 0.97647 |
| | B_Cameras | 0.97346 | 0.97332 |
| | B_Controller Hu. | 0.98014 | 0.98020 |
| | B_Energy Man. | 0.98357 | 0.98370 |
| | B_Health Mon. | 0.98115 | 0.98125 |
| LDA | B_Appliances | 0.82264 | 0.78445 |
| | B_Cameras | 0.77663 | 0.71247 |
| | B_Controller Hu. | 0.84419 | 0.81548 |
| | B_Energy Man. | 0.88020 | 0.86393 |
| | B_Health Mon. | 0.83359 | 0.80042 |
| MLP | B_Appliances | 0.83539 | 0.84229 |
| | B_Cameras | 0.78183 | 0.78121 |
| | B_Controller Hu. | 0.80829 | 0.81048 |
| | B_Energy Man. | 0.84784 | 0.85557 |
| | B_Health Mon. | 0.82085 | 0.82369 |
| QDA | B_Appliances | 0.81880 | 0.78011 |
| | B_Cameras | 0.77285 | 0.70818 |
| | B_Controller Hu. | 0.84035 | 0.81128 |
| | B_Energy Man. | 0.87764 | 0.86139 |
| | B_Health Mon. | 0.83099 | 0.79787 |
| RF | B_Appliances | 0.99971 | 0.99971 |
| | B_Cameras | 0.99976 | 0.99976 |
| | B_Controller Hu. | 0.99987 | 0.99987 |
| | B_Energy Man. | 0.99987 | 0.99987 |
| | B_Health Mon. | 0.99986 | 0.99986 |
| XGB | B_Appliances | 0.99919 | 0.99919 |
| | B_Cameras | 0.99969 | 0.99969 |
| | B_Controller Hu. | 0.99983 | 0.99983 |
| | B_Energy Man. | 0.99988 | 0.99988 |
| | B_Health Mon. | 0.99986 | 0.99986 |

traffic is related to its category, and the classification model should consider this. However, even with this difference in the network traffic pattern, some algorithms could detect the new traffic with an accuracy higher than 98%.

The only algorithm that didn't achieve a decrease or increase in any of the tests was the GNB. In other words, adding examples of a new IoT device category did not influence the algorithm's performance. In this case, it would be interesting to investigate the misclassified samples or even use a multiclass model to better understand this behavior.

Table 5: Experiment 2 - Adding a new smart home IoT device (training set does not contain flows of the specified IoT type).

| Algorithm | Training case | F1 | Accuracy |
|---|---|---|---|
| ADA | Appliances | 0.99572 | 0.99565 |
| | Cameras | 0.92951 | 0.92275 |
| | Controller Hu. | 0.98771 | 0.98691 |
| | Energy Man. | 0.99990 | 0.99990 |
| | Health Mon. | 0.99989 | 0.99989 |
| DT | Appliances | 0.99949 | 0.99949 |
| | Cameras | 0.94372 | 0.93861 |
| | Controller Hu. | 0.99990 | 0.99990 |
| | Energy Mana. | 0.99985 | 0.99985 |
| | Health Mon. | 0.99970 | 0.99970 |
| GB | Appliances | 0.99903 | 0.99903 |
| | Cameras | 0.92879 | 0.92196 |
| | Controller Hu. | 0.99991 | 0.99991 |
| | Energy Man. | 0.99997 | 0.99997 |
| | Health Mon. | 0.99983 | 0.99983 |
| GNB | Appliances | 0.82142 | 0.78259 |
| | Cameras | 0.77562 | 0.71070 |
| | Controller Hu. | 0.84315 | 0.81397 |
| | Energy Man. | 0.88022 | 0.86391 |
| | Health Mon. | 0.83360 | 0.80037 |
| KNN | Appliances | 0.84299 | 0.82320 |
| | Cameras | 0.81461 | 0.78572 |
| | Controller Hu. | 0.86175 | 0.84635 |
| | Energy Man. | 0.94459 | 0.94156 |
| | Health Mon. | 0.91485 | 0.90595 |
| LDA | Appliances | 0.80994 | 0.76943 |
| | Cameras | 0.77869 | 0.71951 |
| | Controller Hu. | 0.85269 | 0.83258 |
| | Energy Man. | 0.87136 | 0.85561 |
| | Health Mon. | 0.83499 | 0.80328 |
| MLP | Appliances | 0.77039 | 0.79052 |
| | Cameras | 0.66707 | 0.69341 |
| | Controller Hu. | 0.76468 | 0.77771 |
| | Energy Man. | 0.83036 | 0.84034 |
| | Health Mon. | 0.80386 | 0.81561 |
| QDA | Appliances | 0.82226 | 0.78485 |
| | Cameras | 0.77342 | 0.70869 |
| | Controller Hu. | 0.84065 | 0.81240 |
| | Energy Man. | 0.87750 | 0.86127 |
| | Health Mon. | 0.83127 | 0.79822 |
| RF | Appliances | 0.99925 | 0.99925 |
| | Cameras | 0.93045 | 0.92387 |
| | Controller Hu. | 0.98985 | 0.98931 |
| | Energy Man. | 0.99993 | 0.99993 |
| | Health Mon. | 0.99986 | 0.99986 |
| XGB | Appliances | 0.99927 | 0.99927 |
| | Cameras | 0.93130 | 0.92489 |
| | Controller Hu. | 0.98817 | 0.98744 |
| | Energy Man. | 0.99991 | 0.99991 |
| | Health Mon. | 0.99818 | 0.99816 |

## 5 CONCLUSION

Traffic classification is a problem that can help improve the security and quality of service aspects of home networks. This paper presented a study considering two types of IoT and non-IoT (mobile and PCs). We evaluated the performance of supervised models in two tasks: learning the impact of different non-IoT devices on traffic classification and ii) identifying a new IoT device type. Our results suggest that models trained with mixed non-IoT traffic benefit the classifier. Despite the particularities of some IoT device traffic, some algorithms could identify a new IoT device with reasonable accuracy. We intend to apply the results in the anomaly detection problem and analyze attack flows on different types of devices for future work.

## REFERENCES


Ahmed, M., Naser Mahmood, A., and Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19–31.

Al Khater, N. and Overill, R. E. (2016). Network traffic classification techniques and challenges. *The 10th International Conference on Digital Information Management*, (Icdim):43–48.

Apthorpe, N., Reisman, D., and Feamster, N. (2017). A smart home is no castle: Privacy vulnerabilities of encrypted iot traffic. *arXiv preprint arXiv:1705.06805*.

Cherif, I. L. and Kortebi, A. (2019). On using eXtreme Gradient Boosting (XGBoost) Machine Learning algorithm for Home Network Traffic Classification. *IFIP Wireless Days*, pages 1–6.

Cvitić, I., Peraković, D., Periša, M., and Gupta, B. (2021). Ensemble machine learning approach for classification of IoT devices in smart home. *International Journal of Machine Learning and Cybernetics*, (0123456789).

Draper-Gil, G., Lashkari, A. H., Mamun, M. S. I., and Ghorbani, A. A. (2016). Characterization of encrypted and VPN traffic using time-related features. *Proceedings of the 2nd International Conference on Information Systems Security and Privacy*, pages 407–414.

Holbrook, L. and Alamaniotis, M. (2019). Internet of things security analytics and solutions with deep learning. In *Proceedings of the 31st IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2019)*, pages 178–185.

Karagiannis, T., Broido, A., Faloutsos, M., and Claffy, K. (2004). Transport layer identification of P2P traffic. *Proceedings of the 2004 ACM SIGCOMM Internet Measurement Conference*, pages 121–134.

Kurose, J. F. and Ross, K. W. (2016). *Computer Networking: A Top-Down Approach*. Pearson, 7th edition.

Lashkari, A. H., Gil, G. D., Mamun, M. S. I., and Ghorbani, A. A. (2017). Characterization of tor traffic using time based features. In *Proceedings of the 3rd International Conference on Information Systems Security and Privacy*, pages 253–262.

Lashkari, A. H., Kadir, A. F. A., Taheri, L., and Ghorbani, A. A. (2018). Toward developing a systematic approach to generate benchmark android malware datasets and classification. In *Proceedings of the 2018 International Carnahan Conference on Security Technology*, pages 1–7.

Liu, Y., Wang, J., Li, J., Niu, S., and Song, H. (2021). Machine Learning for the Detection and Identification of Internet of Things (IoT) Devices: A Survey. *IEEE Internet of Things Journal*, 7(5):1–23.

Ren, X., Gu, H., and Wei, W. (2021). Tree-RNN: Tree structural recurrent neural network for network traffic classification. *Expert Systems with Applications*, 167:114363.

Shahid, M. R., Blanc, G., Zhang, Z., and Debar, H. (2018). IoT Devices Recognition Through Network Traffic Analysis. *Proceedings of the 2018 IEEE International Conference on Big Data*, pages 5187–5192.

Sivanathan, A., Gharakheili, H. H., Loi, F., Radford, A., Wijenayake, C., Vishwanath, A., and Sivaraman, V. (2019). Classifying IoT devices in smart environments using network traffic characteristics. *IEEE Transactions on Mobile Computing*, 18(8):1745–1759.

Soysal, M. and Schmidt, E. G. (2010). Machine learning algorithms for accurate flow-based network traffic classification: Evaluation and comparison. *Performance Evaluation*, 67(6):451–467.

Sperotto, A., Schaffrath, G., Sadre, R., Morariu, C., Pras, A., and Stiller, B. (2010). An overview of ip flow-based intrusion detection. *IEEE communications surveys & tutorials*, 12(3):343–356.

Tahaei, H., Afifi, F., Asemi, A., Zaki, F., and Anuar, N. B. (2020). The rise of traffic classification in IoT networks: A survey. *Journal of Network and Computer Applications*, 154.