International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 3, June 2023, pp. 3449~3463
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i3.pp3449-3463

Machine learning for internet of things classification using network traffic parameters

Loubna Elhaloui[1], Sanaa El Filali[1], El Habib Benlahmer[1], Mohamed Tabaa[2], Youness Tace[1,2], Nouha Rida[3]
[1]Laboratory of Information Technologies and Modelling, Faculty of Sciences Ben M'sik, Hassan II University, Casablanca, Morocco
[2]Pluridisciplinary Laboratory of Research and Innovation (LPRI), EMSI Casablanca, Casablanca, Morocco
[3]Department of Computer Science Engineering, Mohammadia School of Engineers (EMI), Rabat, Morocco

Article Info

Article history:

Received May 24, 2022
Revised Jul 26, 2022
Accepted Aug 18, 2022

Keywords:

Internet of things
Machine learning
Network traffic

ABSTRACT

With the growth of the internet of things (IoT) smart objects, managing these objects becomes a very important challenge, to know the total number of interconnected objects on a heterogeneous network, and if they are functioning correctly; the use of IoT objects can have advantages in terms of comfort, efficiency, and cost. In this context, the identification of IoT objects is the first step to help owners manage them and ensure the security of their IoT environments such as smart homes, smart buildings, or smart cities. In this paper, to meet the need for IoT object identification, we have deployed an intelligent environment to collect all network traffic traces based on a diverse list of IoT in real-time conditions. In the exploratory phase of this traffic, we have developed learning models capable of identifying and classifying connected IoT objects in our environment. We have applied the six supervised machine learning algorithms: support vector machine, decision tree (DT), random forest (RF), k-nearest neighbors, naive Bayes, and stochastic gradient descent classifier. Finally, the experimental results indicate that the DT and RF models proved to be the most effective and demonstrate an accuracy of 97.72% on the analysis of network traffic data and more particularly information contained in network protocols. Most IoT objects are identified and classified with an accuracy of 99.21%.

Corresponding Author:

Loubna Elhaloui
Laboratory of Information Technology and Modeling, Faculty of Sciences Ben M'sik, Hassan II University of Casablanca
BP 7955 Sidi Othman Casablanca, Morocco
Email: l.elhaloui@gmail.com

1. INTRODUCTION

Nowadays, the telecommunications market is experiencing a significant boom in the use of smart connected objects. This object is a hardware component equipped with a sensor that allows data to be generated, exchanged, and consumed with minimal human intervention [1]. They have an increasingly important presence in our daily life, whether in our ways of consuming or in our ways of producing. In particular, these smart objects make it possible to create a mass of available data, thanks to the collection and processing of the traffic sent and received by each connected object on an IoT network, to make our environment smarter, in particular, smart homes, smart buildings, smart traffic, and smart cities [2].

In our previous work [3], we presented the IoT system model of a smart building, to allow users to control, identify and access smart devices, thanks to the shared and exchanged data by different network protocols. It, therefore, becomes necessary to be able to secure these various objects. The identification of the

intelligent objects which evolves in a network constitutes is an essential component of the network management tools because it provides important information allowing, in particular, to ensure the legitimacy of the traffic exchanged.

Unfortunately, this modeling has demonstrated limitations related to the detection of physical objects connected in a heterogeneous network. The main limitation is that all objects cannot be detected through a single gateway due to a variety of IoT protocols. Recently, some researchers have presented techniques for identifying IoT objects that rely on learning methods to characterize the attributes of various objects. Sivanathan *et al.* [4] developed an algorithm for classifying IoT devices based on machine learning, which is based on various network traffic characteristics to identify and classify the behavior of IoT objects on a network. Ammar *et al.* [5] used supervised learning techniques based on flow attributes of traffic sent and received by connected objects as well as textual data. Meidan *et al*. [6] are the first to demonstrate the feasibility of identifying IoT objects based on network traces using machine learning. In the first step, a system that analyzes TCP sessions is presented to differentiate network traffic generated by non-IoT and IoT objects, and in the second step, their identification is proceed. Snehi and Bhandari [7] proposed a new framework for IoT traffic classification based on Stack-Ensemble, by exploiting the behavioral attributes of real-time high-volume IoT device traffic. Bezawada *et al.* [8] proposed a complementary identification system that leads to the behavioral identification of IoT objects based on their activity within the network. In addition, Miettinen *et al*. [9] presented a system for automatically identifying IoT objects and enforcing security that executes an appropriate action plan to restrict or authorize their communications within a network. Sneh and Bhandari [10] provided the taxonomy of the techno functional application domains of the IoT classification, by inferences on the attributes of IoT traffic and the exploitation of an Australian dataset collected from 28 IoT objects.

In this paper, we present an implementation of a model for classifying connected objects by an identification system through network protocols and traffic flow statistics, using the packet analysis tools executed in the gateway (to see all incoming and outgoing traffic from connected objects). The discipline of traffic flow analysis provides a means of collecting and exporting data that infer attributes of packets.

This article is organized as follows. Section 2 describes the problem of the work citing relevant previous work. Presenting the literature concerning machine learning algorithms with the state of the art in section 3. IoT traffic parameters in section 4, and in section 5 develop classification models to identify IoT objects. The paper is concluded in section 6.

2. BACKGROUND

The growing number of devices connected to the internet capable of communicating with each other continues to increase at a steady pace [2]. This trend tends to increase with the proliferation of actors, both manufacturers and suppliers. The IoT based on traditional networks to which so-called “intelligent” objects are connected, raises new issues around the detection of connected objects on heterogeneous networks involved in intelligent environments, and also around the security [11] of these networks and the information passing through them.

The identification of connected objects poses a great challenge given a large number of heterogeneous protocols [12], the networks used and few consensual standards. Recent approaches to object identification based on behavioral analysis of computing devices have emerged [13]. The basic idea is to scrutinize the traffic crossing the network, using either active or passive measurement techniques, and to extract unique patterns that are sufficiently discriminating in order to individually identify the objects present within our network. There are a wide variety of methods for analyzing device traffic flow, that can be broadly classified into two categories depending on the type of network surveillance considered: active surveillance or passive surveillance.

The principle of active surveillance is to generate traffic in the network and observe any reactions to the stimulus. As such, it creates additional traffic in the network. Conversely, in the case of passive surveillance, it is an approach considered less intrusive, consisting in capturing the traffic crossing the network and studying its properties at one or more points of the network. Usually, this approach requires software tools for traffic capture or analysis like Wireshark [14], tcpdump, NetworkMiner, and WinDump.

Sivanathan *et al.* [15] have conducted tests to determine the feasibility of identifying the type of an IoT device by probing its open ports. Nmap [16] is used to scan the ports of 19 IoT devices from their test bench, in order to build a knowledge base of IoT device port number combinations thus forming their signature. Snehi and Bhandari [7] have proposed a new Stack-Ensemble framework for IoT traffic classification that characterizes traffic ingress based on statistical and functional attributes of IoT devices. This proposed framework is capable of managing network traffic in real time. The authors have performed a comparative analysis between the stack-Ensemble model and other classification models such as XGBoost stacks, distributed random forest, gradient boosting machine, and general linear machine algorithms.

Through this analysis, their framework demonstrated the highest values of accuracy compared to other classification models.

Miettinen *et al.* [9] proposed a system called IoT sentinel which identifies types of IoT devices and executes an appropriate course of action to restrict or allow their communications within a network. So that any device, or attack vectors, are not used to compromise the entire network. The system relies on the random forest classification model to identify the type of object. According to the authors, two devices are said to be of the same type if they share the same model and the same software version. When a new device is introduced into the network for the first time, when a new MAC address is discovered, and then the latter begins its installation and configuration phase (first moments of communication with the gateway). In this case, the system initiates a packet capture process using tcpdump with filtering by the MAC address of the new device. Bezawada *et al.* [8] propose a complementary system called IoTSense which performs behavioral identification of IoT devices based on their activity within the network by analyzing ethernet, IP, and transport headers. Each device is assigned a behavioral profile, so as to detect possible deviations from the initial behavior of the device, due to malicious activities for example. The abbreviations used in the literature are defined in Table 1.

Table 1. Abbreviations used in the literature

Abbreviation	Description
ARP	Address resolution protocol
DNS	Domain name system
DRF	Distributed random forest
DT	Decision tree
EMSI	Moroccan School of Engineering Sciences
GBM	Gradient boost machine
GLM	Generalized linear model
HTTP	Hypertext transfer protocol
HTTPS	Hypertext transfer protocol secure
ICMP	Internet control message protocol
IoT	Internet of things
IP	Internet protocol
KNN	K-nearest neighbors
LPRI	Multidisciplinary Research and Innovation Laboratory
MAC	Media access control
MDNS	Multicast domain name system
ML	Machine learning
NB	Naive Bayes
NTP	Network time protocol
RF	Random forest
SGDC	Stochastic gradient descent classifier
SSDP	Simple service discovery protocol
SSL	Secure socket layer
SVM	Support vector machine
TCP	Transmission control protocol
TLS	Transport layer security
UDP	User datagram protocol

3. MACHINE LEARNING: STATE-OF-ART

Machine learning is part of one of the approaches to artificial intelligence [17], which consists of creating algorithms capable of improving automatically with experience. It is increasingly integrated into most of the technologies we use on a daily basis. The machine “learns” prior data and adapts its responses. Utilizing machine learning involves using datasets of different sizes to identify correlations, similarities, and differences [18].

Furthermore, ML makes extensive use of tools and concepts from statistics and is part of a larger discipline called “data science”. There are three main types of ML, Supervised learning aims to establish rules of behavior from a dataset containing examples of already labeled cases [19]. Unsupervised learning, unlike supervised learning; unsupervised learning deals with the case where we only have the inputs, without first having the outputs. The goal of unsupervised learning is to find hidden shapes in an unlabeled dataset [19]. Reinforcement learning is a type of ML in which a model has no training data at the start. The objective is for an agent to evolve in an environment and learn from its own experience. For a reinforcement learning algorithm to work, the environment in which it operates must be computable and have a function that evaluates the quality of an agent [19].

The identification of IoT objects presented in this work is based on supervised learning. More precisely, it is treated as a supervised classification problem. To this end, we focus on six classification algorithms, their finer details on each model are given as follows.

3.1. Support vector machine

SVM are algorithms that separate data based on classes or separators [20]. The SVM algorithm is ideal for identifying simple classes which are separated by vectors called hyperplanes, and which distinguish data based on training class labels. It is also possible to program the algorithm for nonlinear data, which cannot be clearly separated by vectors. Principally, an SVM is all about finding the hyperplanes that best separate data classes. The predicted classes in model SVM are made based on the side of the hyperplane where the data point falls. SVM is a kind of supervised learning algorithm based on structural risk minimization [21]. As a popular machine learning algorithm, SVM is widely used in many fields, such as finance and information retrieval, it provides high accuracy on current and future data. The functional part of the solution to the SVM problem is written as a linear combination of the kernel functions taken at the support points:

$$f(x) = \sum_{i \in A} \alpha_i y_i k(x, x_i)$$

where *A* denotes the set of active constraints and the *αi* the solutions of the following quadratic program:

$$\begin{cases} \min\limits_{\alpha \in \mathbb{R}^n} \dfrac{1}{2} \alpha^T G \alpha - e^T \alpha \\ avec \quad y^T \alpha = 0 \\ 0 \leq \alpha_i \leq C \end{cases}$$

where *G* is the matrix $n \times n$ with general term $G_{ij} = y_i y_j k(x_i, x_j)$. The bias b to the value of the Lagrange multiplier of the equality constraint at the optimum [22].

3.2. Decision tree

A DT is a supervised learning algorithm primarily used to graph data in branches to show possible outcomes of various actions. Classification and prediction use response variables based on past decisions [23]. DT forms a flowchart like a tree, where each node represents the test on the attribute, and each branch denotes the result of the test. The leaf node owns the class label. However, decision trees become difficult to read when associated with large volumes of data and complex variables. A DT is a type of learning algorithm that can be applied to many contexts: finance, pharmaceuticals, and agriculture.

In the case of classification, the classification and regression trees (CART) algorithm uses the Gini diversity index to measure the classification error [24]. Practically, if we suppose that the class takes a value in the set *1, 2, ..., m*, and if fi denotes the fraction of the elements of the set with label i in the set, we have:

$$I_G(f) = \sum_{i=1}^{m} f_i(1 - f_i) = \sum_{i=1}^{m} (f_i - f_i^2) = \sum_{i=1}^{m} f_i - \sum_{i=1}^{m} f_i^2 = 1 - \sum_{i=1}^{m} f_i^2 \, v$$

3.3. Random forest

RF is a supervised learning technique that uses ensemble learning algorithms that combines an aggregation technique, “Bagging”, and a particular decision tree induction technique. It creates a strong classifier based on weak classifiers [25]. As the name suggests, RFs are formed by simply assembling multiple decision trees, usually ranging from a few tens to thousands of trees. This bagging method forms patterns, which are responsible for increased performance [21]. In addition, the random process in the construction of the trees makes it possible to ensure a low correlation between them. RF is also known for its accuracy and ability to process datasets composed of few observations and many features. It is used in crop classification and prediction of crop yield corresponding to current climatic and biophysical changes [26].

Let $\hat{h}(., \theta_1), \ldots, \hat{h}(., \theta_q)$ be a collection of tree predictors, with $\theta_1, \ldots, \theta_q$ q random variables i.i.d. independent of L_n [27]. The RF predictor $\hat{h}_{RF}$ is obtained by aggregating this collection of random trees as follows.

$$\hat{h}_{RF}(x) = \frac{1}{q} \sum_{l=1}^{q} \hat{h}(x_1, \theta_l)$$

$$\hat{h}_{RF}(x) = arg\, max \sum_{l=1}^{q} \mathbb{I}_{\hat{h}(x_1, \theta_l)} = k \quad avec\ 1 \leq k \leq K$$

3.4. K-nearest neighbor

KNN is a supervised learning method [28]. It is used for regression and classification. To make a prediction, the KNN will be based on the datasets. The datasets are trained according to their class. The KNN algorithm needs a distance calculation function between observations, it must be predicted to calculate the distance with the nearest "K" points [21]. Using the formulas, there are several distance calculation methods including, Minkowski distance, Manhattan distance, Euclidean distance, and Hamming distance. We choose the distance method according to the types of data we are handling. The choice of the highest number of K to make a prediction with the KNN algorithm, varies depending on the dataset. In agriculture, the KNN is very effective for the classification of different cereals-cultivars of cereals [21]. There are different distance calculations used in the comparison step of the KNN algorithm such as:

a. Euclidean distance, which has been used in several identification systems based on the KNN algorithm [29]. The Euclidean distance $d_E(X,Y)$ between the two vectors X and Y is given by

$$d_E(X,Y) = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2}$$

b. Distance from city block, which is defined as follows.

$$d_E(X,Y) = \sum_{i=1}^{m}|x_i - y_i|$$

c. Cosine distance, which is also called angular distance and is derived from cosine similarity which measures the angle between two vectors. This distance is defined as follows.

$$d_{cos}(X,Y) = 1 - \frac{\sum_{i=1}^{m} X_i Y_i}{\sqrt{\sum_{i=1}^{m} X_i^2}} \frac{1}{\sqrt{\sum_{i=1}^{m} Y_i^2}}$$

d. Correlation distance, which is given by the following formula.

$$d_{cor}(X,Y) = 1 - \frac{\sum_{i=1}^{m}(x_i - \bar{y}_i)}{\sqrt{\sum_{i=1}^{m}(x_i - \bar{y}_i)^2}} \frac{(x_i - \bar{y}_i)}{\sqrt{\sum_{i=1}^{m}(x_i - \bar{y}_i)^2}}$$

3.5. Naive Bayes classifier

NB classifier is a supervised machine learning algorithm [30], it is a classification method that is mainly based on Bayes' theorem. The latter is particularly useful for text classification issues. Bayes' theorem is based on conditional probability theory [31]. The NB algorithm defines rules that allow it to classify a set of observations, thus defining its classification rules from a dataset in order to apply them to the classification of predictive data. Its main function is that it makes a strong priori hypothesis of the independence of the characteristics considered, thus ignoring the correlations that may exist between them. NB algorithms are widely used in the creation of Anti-Spam filters, recommendation systems, and digital marketing. The probabilistic model for a classifier is the conditional model [32].

$$p(C|F_1, \dots, F_n)$$

where C is a dependent class variable whose instances or classes are few, conditioned by serval characteristic variables $F_1, \dots, F_n$. Using Bayes' theorem, we write:

$$p(C|F_1, \dots, F_n) = \frac{p(C)p(F_1, \dots, F_n|C)}{p(F_1, \dots, F_n)}$$

3.6. Stochastic gradient descent classifier

Stochastic gradient descent classifier (SGDC) is a supervised predictive learning algorithm [33], which will allow to minimize the objective function which is written as a sum of differentiable functions. The process is then performed iteratively on randomly drawn datasets. Each objective function minimized in this

way is an approximation of the global objective function. The SGDC is widely used for training many families of models in machine learning, including support vector machines, logistic regression and graphical models [34]. In the SGD algorithm, the true value of the gradient of $Q(w)$ is approximated by the gradient of a single component of the sum.

$$Q(w) = \frac{1}{n}\sum_{i=1}^{n} Q_i(w)$$

In pseudo-code, the SGD method can be represented.

a. Choose an initial parameter vector *w* and a learning rate η.
b. Repeat until an approximate minimum is obtained: i) randomly shuffle the samples from the training set, and ii) for $i = 1, 2, \ldots, n$, do:

$$w := w - \eta \nabla Q(w) = w - \frac{\eta}{n}\sum_{i=1}^{n} \nabla Q_i(w)$$

4. INTERNET OF THINGS TRAFFIC PARAMETERS

Understanding the nature and characteristics of the traffic generated by IoT objects is a crucial step for implementing effective network policy and resource management in an IoT infrastructure. However, studies focusing exclusively on characterizing IoT traffic are still in their infancy. A challenge that Sivanathan *et al*. [35] attempted to address by empirically analyzing network traffic under conditions simulating a smart city and smart campus environment in order to uncover the characteristics and behavioral patterns of IoT devices.

To do this, they collected network traffic from a heterogeneous range of 30 devices, both 28 IoT devices and 2 non-IoT devices, over a continuous period spanning several months. IoT traffic includes both traffic generated by devices autonomously and traffic generated as a result of user interactions with devices. The raw data collected consists of the TCP packet data header and payload information.

The authors are primarily interested in the distribution of 4 traffic flow characteristics: duration, ratio, throughput, and the duration of inactivity of traffic flows. It is explained that for each of the characteristics there are disparities that exhibit a distinct pattern. Sivanathan *et al.* [35] explained that each of the IoT devices uses less than 10 distinct ports to communicate and that some devices use non-standard port numbers. Moreover, some of them from the same manufacturer share some port numbers. Similarly, in terms of DNS queries, certain domain names are invoked by devices from the same manufacturer. The authors have also pointed out that with respect to the NTP protocol, some devices exhibit an identifiable pattern at the NTP request sending interval. Finally, they noticed that 17 of the 28 IoT devices in the test bed use TLS/SSL to communicate. Also, at the list of cipher suites [36] issued when establishing a TLS connection.

In our object identification process based on machine learning techniques, we have conducted tests to determine the feasibility of detecting smart objects by probing their network traffic. We have used Wireshark [14] to scan our network traffic of 75 devices, in order to build a knowledge base of combinations of IP addresses, MAC addresses, port numbers, and packet sizes. Firstly, we have analyzed the protocol sessions to distinguish the network traffic generated by the IoT objects, and secondly, to proceed to their identification. Our work describes an experimental environment in which network traffic data was collected from 75 objects of 13 different types of devices. Over a period of several months, traffic capture was recorded as packets in PCAP files. This collected data is then transformed into protocol sessions (ARP, SSDP, mDNS, DNS, NTP, HTTP, HTTPS, TCP, and UDP), each session is identified by a unique triplet (source address, destination address, type of protocol).

In this study, using supervised learning, classification models such as the RF model, DT, and KNN model were used. to train a classifier that predicts the probability that a given session originates from an object belonging to the set of known IoT objects. Initially, the results show an average rate of 89% of sessions correctly classified as being part of our list of objects. Then to improve these results, we have put an additional step in the classification process using the balancing on the network traffic coming from each connected object in our environment. The result shows an improvement of around 8%. In this regard, we have chosen the classification models of supervised machine learning, to proceed with the identification of IoT objects. Due to the heterogeneity of the protocols and devices of the latter, the classification model which presents a rate of 97.72% is the decision tree.

5. DISCUSSION

A smart building uses technology to share information between different systems [37], it is happening in the building in order to optimize the performance of the latter. This information is then used to automate various processes, from heating to ventilation, or air conditioning for security. When we talk about smart buildings, the general public thinks first of all, of a building that intelligently monitors its energy consumption and is able to control this consumption, because it relies above all on connectivity. It is made up of connected objects and applications with which the user interacts in real time. But the concept is much broader than that. A smart building also has advantages in the areas of living comfort, health and safety, among others [38].

The most fundamental characteristics of the smart building are its systems that are connected to each other. This system consists of smart objects, such as fire alarms, lighting, motion detectors, cameras; they are all connected. The use of smart objects is an integral part of a smart building, and they play a very important role in collecting data for collection and analysis by automated systems that can identify and control throughout the building. In the present work, the IoT environment is discussed through the prism of connected objects evolving in a similar intelligent building has been set up within the framework of the LPRI as shown in Figure 1 at EMSI, one involving IoT devices in Table 2, gas sensors, cameras, smart speakers, temperature sensors, IP phones, smart TVs, smartphones are connected to the internet.

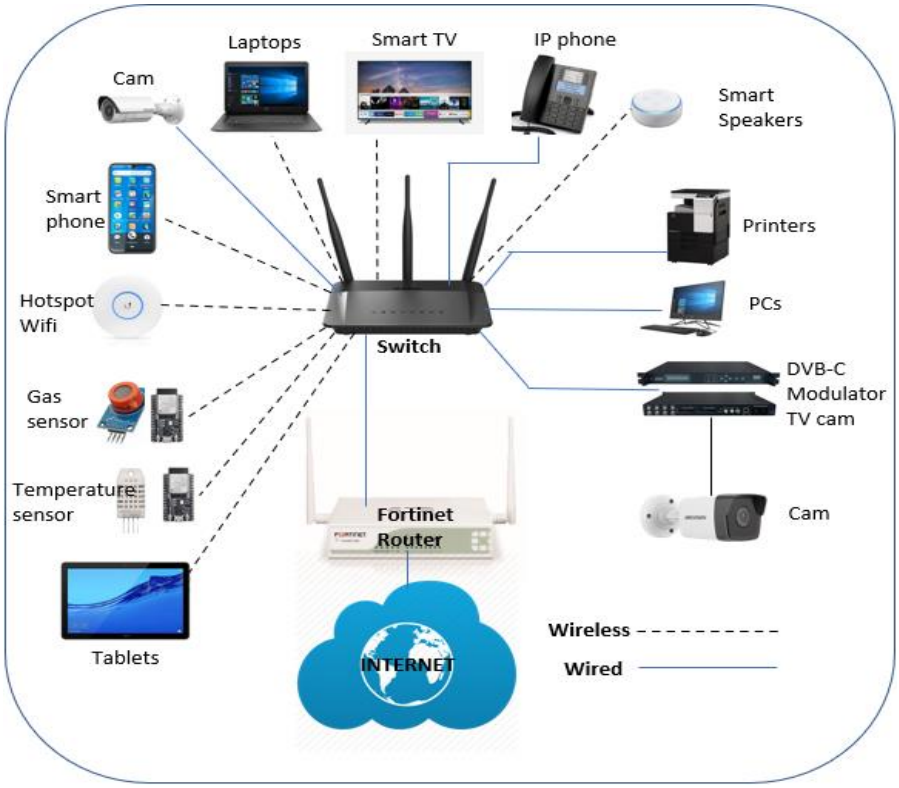


Figure 1. Architecture of the IoT environment of the LPRI lab

Table 2. List of devices used in our lab

Devices	Number of devices	Number of flows generated
Smart TV (Samsung)	4	36793
Printer (Tokyo Electric CO., LTD)	3	37154
Smart Speaker (JBL)	4	36473
WebCam (Hangzhou Hikvision)	9	37429
Hotspot WIFI (Ubiquiti Access Point)	6	37940
Gas Sensor	4	36396
Temperature Sensor	3	33150
Smart Phone	6	32454
Laptop	6	36328
Personal Computer	10	37944
IP Phone (Aastra)	10	34048
Modulator DVB-C	4	37333
Tablet (Samsung)	6	34412

On the other hand, studies focusing exclusively on characterizing IoT traffic are still in their infancy. To do our job, we collected network traffic from a diverse range of devices, over a continuous period of time spanning multiple times. IoT traffic includes both traffic generated by devices autonomously and traffic generated as a result of user interactions with devices.

Figure 2 represents the operating principle of our system for identifying IoT objects in our environment, starting from the capture of network traffic to the development of classification and prediction models. Firstly, this system collects the network traffic from the start of the object to be identified. Then, a step of extracting parameters characterizing the different classes is carried out from the traces of IoT traffic. The next step is to classify all the extracted parameters to obtain the identity of the considered object using one or more classifiers such as SVM, KNN, RF, and DT. This classification takes into account the models of the different classes, previously trained in a phase called the learning phase.

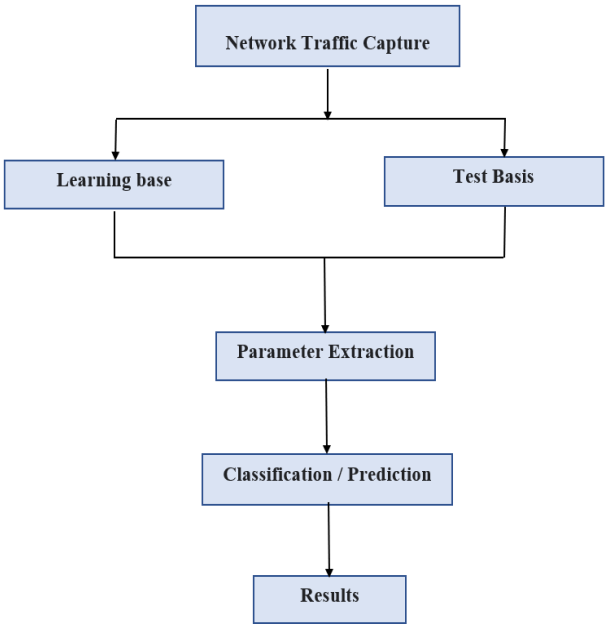


Figure 2. General view of the operation of an identification system

The raw data collected consists of the TCP packet data header and payload information. We are first interested in the distribution of traffic flow characteristics such as throughput, duration, and idle time of traffic flows. We will explain that for each of the characteristics where we find disparities that exhibit a distinct pattern.

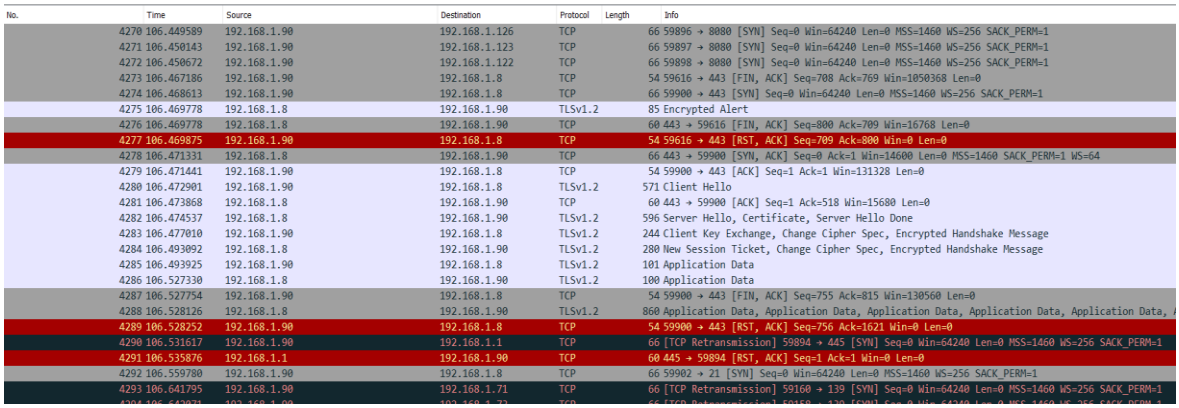
Capturing network traffic is a relatively easy process that can be accomplished by placing a tool such as Wireshark or t-shark on a host through which network traffic is routed. In our case, all network traffic entering and leaving the local network was observed and collected manually using the Wireshark tool as in Figure 3. During this observation phase, all traces were collected several times from a computer (Microsoft Windows 10) connected to the same network. The distribution of packet volume per IoT object generally shows variations in magnitude when there are no interactions with third parties. Figure 4 illustrates the distribution of packets of IoT objects in our lab. In particular, we can notice the absence of network activity with regard to the gas sensor and the temperature sensor. However, if one interacts with these latter sensors, then their network activity is multiplied by a variable factor.

We have described the data collection process. Once we have all the traces, we need to convert them into a format usable by the machine learning algorithms. To do this, a python script has been implemented to allow the extraction of the characteristics from the network flow. A network stream can be defined as one or more packets traveling between two computer addresses using a particular protocol (TCP, UDP, ICMP, ...).

Most IoT objects regularly exchange traffic with servers that are often identifiable by their domain names corresponding to their manufacturers/suppliers. In addition, these exchanges can occur periodically, such as the use of the NTP protocol for time-stamping services, or DNS requests at the initiative of IoT objects. Most IoT objects exhibit a recognizable pattern in the use of certain TCP/IP protocols [35].

After the stage of feature extraction based on PCAP files and their transformation into a dataset, this was processed using the Scikit-learn library to develop models capable of predicting/identifying the type of

IoT objects using the machine learning technique. There are multiple different classification algorithms suitable for a problem like this. Many of them inherently support multiclass data (e.g., NB, decision trees, nearest-K), and for others like the SVM which only supports two classes by definition, there are still several methods for adapting SVMs to multiclass problems [39].



No.	Time	Source	Destination	Protocol	Length	Info
4270	106.449589	192.168.1.90	192.168.1.126	TCP	66	59896 → 8080 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
4271	106.450143	192.168.1.90	192.168.1.123	TCP	66	59897 → 8080 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
4272	106.450672	192.168.1.90	192.168.1.122	TCP	66	59898 → 8080 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
4273	106.467186	192.168.1.90	192.168.1.8	TCP	54	59616 → 443 [FIN, ACK] Seq=708 Ack=769 Win=1050368 Len=0
4274	106.468613	192.168.1.90	192.168.1.8	TCP	66	59900 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
4275	106.469778	192.168.1.8	192.168.1.90	TLSv1.2	85	Encrypted Alert
4276	106.469778	192.168.1.8	192.168.1.90	TCP	60	443 → 59616 [FIN, ACK] Seq=800 Ack=709 Win=16768 Len=0
4277	106.469875	192.168.1.90	192.168.1.8	TCP	54	59616 → 443 [RST, ACK] Seq=709 Ack=800 Win=0 Len=0
4278	106.471331	192.168.1.8	192.168.1.90	TCP	66	443 → 59900 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1 WS=64
4279	106.471441	192.168.1.90	192.168.1.8	TCP	54	59900 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
4280	106.472901	192.168.1.90	192.168.1.8	TLSv1.2	571	Client Hello
4281	106.473868	192.168.1.8	192.168.1.90	TCP	60	443 → 59900 [ACK] Seq=1 Ack=518 Win=15680 Len=0
4282	106.474537	192.168.1.8	192.168.1.90	TLSv1.2	596	Server Hello, Certificate, Server Hello Done
4283	106.477010	192.168.1.90	192.168.1.8	TLSv1.2	244	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
4284	106.493092	192.168.1.8	192.168.1.90	TLSv1.2	280	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
4285	106.493925	192.168.1.90	192.168.1.8	TLSv1.2	101	Application Data
4286	106.527330	192.168.1.8	192.168.1.90	TLSv1.2	100	Application Data
4287	106.527754	192.168.1.90	192.168.1.8	TCP	54	59900 → 443 [FIN, ACK] Seq=755 Ack=815 Win=130560 Len=0
4288	106.528126	192.168.1.8	192.168.1.90	TLSv1.2	860	Application Data, Application Data, Application Data, Application Data,
4289	106.528252	192.168.1.90	192.168.1.8	TCP	54	59900 → 443 [RST, ACK] Seq=756 Ack=1621 Win=0 Len=0
4290	106.531617	192.168.1.90	192.168.1.1	TCP	66	[TCP Retransmission] 59894 → 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
4291	106.535876	192.168.1.1	192.168.1.90	TCP	60	445 → 59894 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
4292	106.559780	192.168.1.90	192.168.1.8	TCP	66	59902 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
4293	106.641795	192.168.1.90	192.168.1.71	TCP	66	[TCP Retransmission] 59160 → 139 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1

Figure 3. Wireshark capture

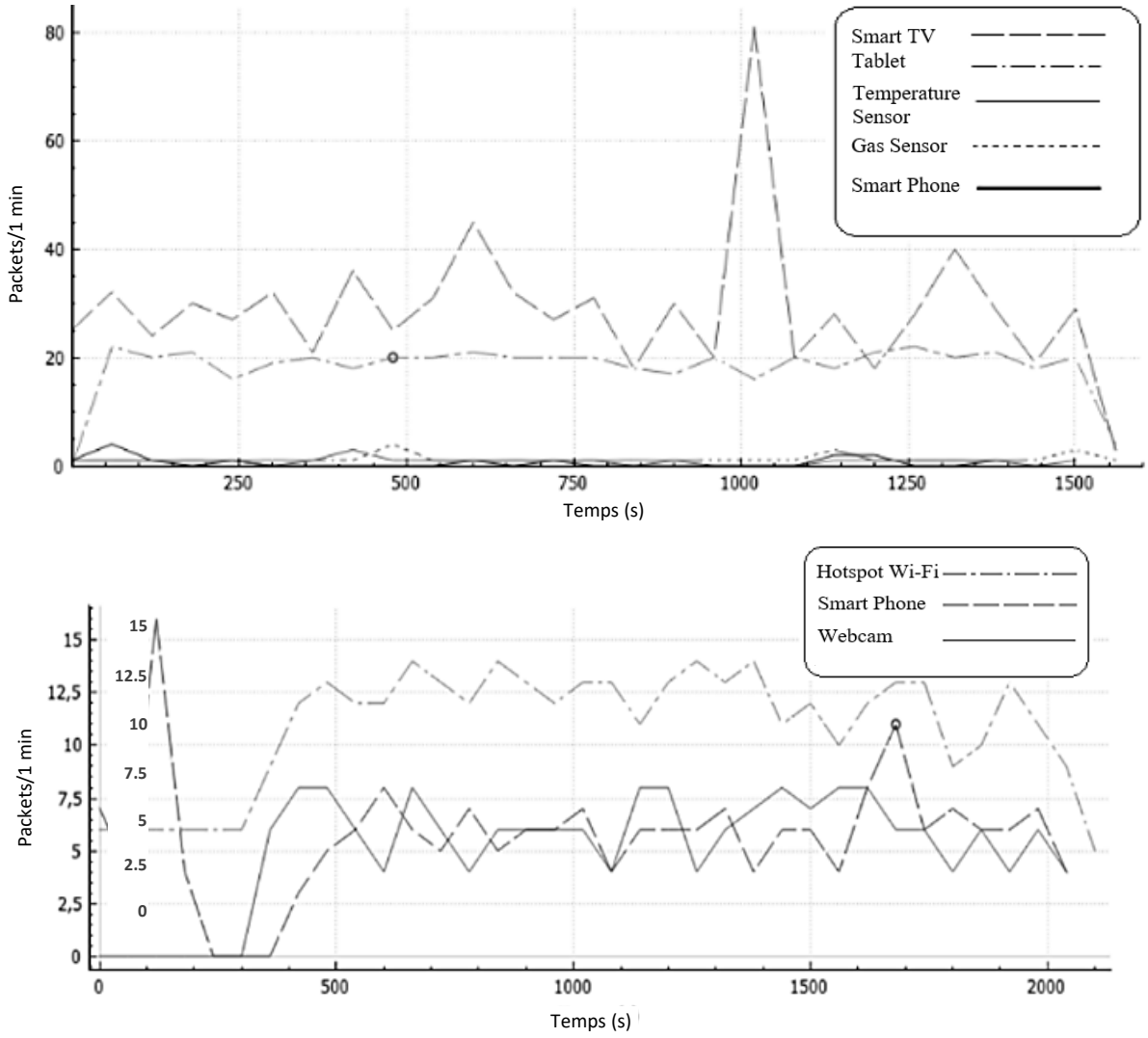


Figure 4. Packet distribution of IoT objects in 1-minute intervals

Scikit-learn includes a wide range of supervised and unsupervised machine learning algorithms. In this work, six different classification algorithms were used: RF, SVM, KNN, SGDC, DT, and NB as in Figure 5. To do this, the algorithms were executed in a web application called Jupyter Notebook [40] chosen for its intelligible interface. As mentioned above, the approach proposed in this paper is based on multiclass supervised learning in the sense that we treat the identification of IoT objects as a supervised classification problem. Our dataset contains a set of values where each value is associated with a feature and an observation.

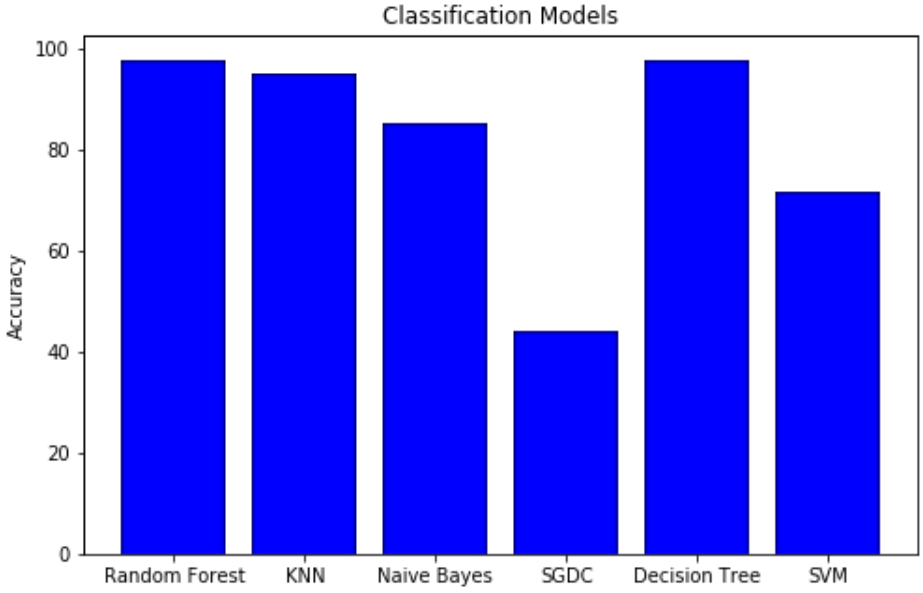


Figure 5. Performance of classification models

Our dataset includes 467,854 observations. It was divided into two subsets (training and test set) during the supervised learning phase. Once the models were trained on the training set, we checked their performance on the test set using metrics from the Scikit-learn library.

Just like on our own dataset, we trained the algorithms on the first subset of data and then evaluated their performance on the second. As a result, the DT and RF models proved to be the most efficient in view of the metric results shown in Figure 4. In addition, their learning time is quite fast compared to others.

To evaluate the performance of the classification of IoT objects, Figures 6 and 7 show the resulting confusion matrices of the two learning algorithms, respectively the decision tree model and the Random Forest model, of this classification. Each given cell of the confusion matrix indicates the precision that receives a positive output from the model in the corresponding row. From the raw outputs of Figures 6 and 7, it can be seen that these two matrices have almost the same values, and all models of the objects correctly detect most instances of their own class, with the exception of objects like hotspot Wi-Fi which have a true positive rate of less than 94%. On the other hand, the other objects show more than 95% up to 100% of correct detection, which is to say true positives, for example, the models of smart TV (Samsung), tablet, and laptop objects have the greatest confidence. At the same time, one can also see the other models incorrectly detecting instances of objects from other classes, i.e., false positives, as shown by the non-diagonal elements in the confusion matrices.

The hotspot Wi-Fi object is more impacted compared to other objects by experiencing a drop in its true positive rate. Focusing on the models of the objects like gas sensor and temperature sensor, we found that their clusters overlapped with each other and with other IoT objects by a certain number of clusters, and therefore they resulted in false positives. We do not forget that these overlaps in the models of IoT objects are expected, especially when we want to classify a large number of different objects. IoT traffic overlaps can be due to various reasons such as actions triggered by events, or the use of common services, such as objects from the same manufacturer.

The final discussion of model performance concerns the details of the critical performance metric (accuracy). Table 3 shows the comparative analysis of the accuracy of IoT objects for the following models DT, RF, and KNN, the higher values of accuracy complement the overall accuracy of each model. Table 4 presents the comparison of the proposed work with state-of-art in the field of IoT classification.

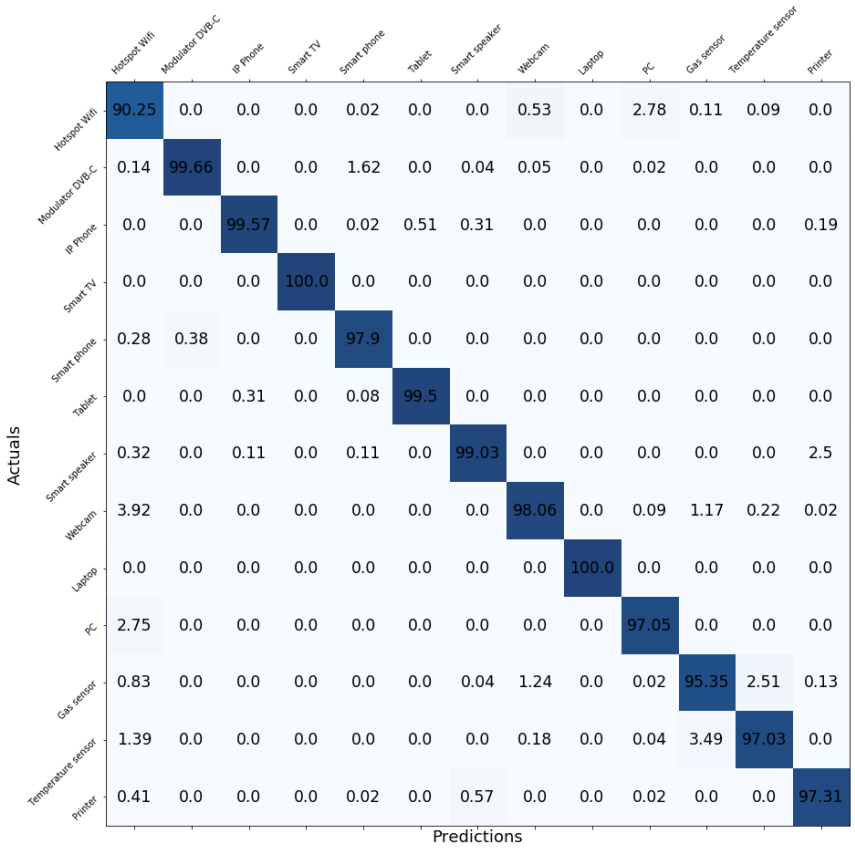


Figure 6. Decision tree model confusion matrix

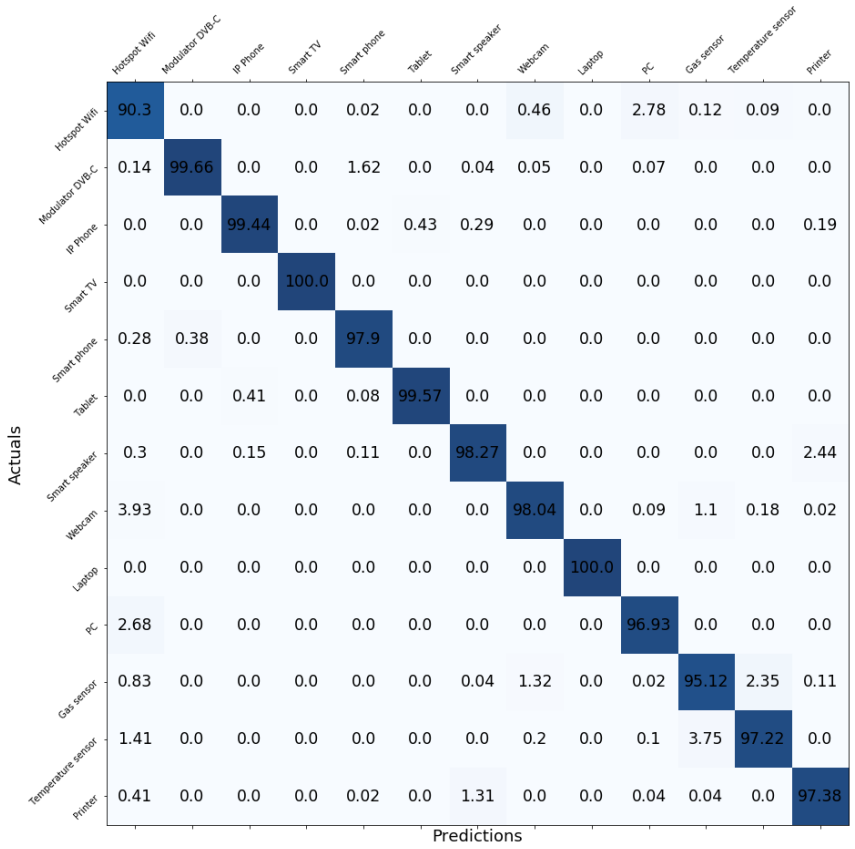


Figure 7. Random forest model confusion matrix

Table 3. Comparison of learning model performance metrics

Devices	Decision Tree	Random Forest	K-Nearest Neighbors
Smart TV (Samsung)	1	1	0.988317757
Smart Speaker (JBL)	0.990300317	0.982652490	0.996662340
Temperature Sensor	0.970267592	0.982652490	0.996662340
Smart Phone	0.979044933	0.979044933	0.953516820
Laptop	1	1	1
IP Phone (Aastra)	0.995714007	0.994350282	1
Modulator DVB-C	0.996612587	0.996612587	1
Tablet (Samsung)	0.994969040	0.995743034	1

Table 4. Comparison of the state of the art in the field of IoT classification

References	Objective	Methods	Testbed Configuration	Performance
[4]	Classifying IoT devices	NB, RF	Smart Lab environment (28 devices)	Port Numbers: Accuracy: 92.13% Domain Names: Accuracy: 79.48% Cipher Suite: Accuracy: 36.15% The final accuracy: 99.88%
[5]	Classification of Connected objects	DT, SVM, NB, RF, KNN	33 connected objects	Traffic flow attributes: accuracy 72% Text attributes: accuracy 93% The accuracy of the DT: 99% The accuracy of the SVM: 88% The accuracy of the NB: 98% The accuracy of the KNN: 94% The accuracy of the RF: 94%
[6]	Classify IoT devices	GBM, eXtreme Gradient Boosting (XGB), RF	9 IoT devices	The total accuracy of the different models used: 99.281%
[7]	IoT/Non-IoT Classification in real-time	Stack-Ensemble, DRF, XGB, GBM, GLM	Packet captures from [4]	The Stack-Ensemble model outperformed with an accuracy of 99.94%
[8]	Fingerprint Classification	KNN, DT, GBM	14 IoT devices	Not specified
[9]	Fingerprint Classification	RF	27 devices	Accuracy: 95%
[41]	IoT Classification	DT	Smart Home setup (5 IoT devices)	Accuracy: 97%
Our proposed work	IoT Classification in real-time	DT, RF, NB, KNN	75 IoT devices from Smart environment (living Lab LPRI in EMSI)	The accuracy of the DT: 97.72% The accuracy of the RF: 97.65% The accuracy of the KNN: 95.15% The accuracy of the NB: 85.09 The final accuracy: 99.21% (80% in all IoT objects)

6. CONCLUSION AND PERSPECTIVES

The main objective of this work was to propose a method for identifying IoT objects by analyzing network traffic data. These were collected and analyzed manually using the Wireshark tool to extract the characteristics of the network flow, which allows us to build our base of exploitable characteristics by learning algorithms. To this end, an infrastructure of connected objects simulating an intelligent environment has been deployed to collect network traffic in real conditions of use.

During the exploratory phase of network traffic, we have developed learning models capable of classifying and identifying connected IoT objects in our work environment. Regarding supervised learning, we subjected our dataset to six different classification algorithms (SVM, KNN, DT, RF, NB, and SGDC). As a result, the DT and RF models proved to be the most efficient in view of the metric results, they achieved 97.72% accuracy in identifying and classifying each IoT object from the IoT dataset (most IoT objects are identified and classified with an accuracy of 99.21%).

Although this approach makes it easier for us to identify and detect smart objects in our environment, it lacks the security of these objects that are connected and interconnected to the internet with its high cybersecurity risk in IoT networks. Currently, the smart environment has increasingly become a target for emerging cyberattacks that will impact user privacy and potential security. In future work, we will study the securing chapter of the IoT, which is a major and important challenge in our daily life, to define the main security problems caused by IoT objects.

REFERENCES

[1] T. Sapkta, "A general survey on internet of things (IoT)," *Indian Journal of Natural Sciences*, vol. 12, no. 66, pp. 32077–32081, 2021.

[2] M. Lombardi, F. Pascale, and D. Santaniello, "Internet of things: A general overview between architectures, protocols and applications," *Information*, vol. 12, no. 2, Feb. 2021, doi: 10.3390/info12020087.

[3] L. Elhaloui, S. Elfilali, M. Tabaa, and E. H. Benlahmer, "Toward a monitoring system based on IoT devices for smart buildings," in *Advances on Smart and Soft Computing*, 2021, pp. 285–293. doi: 10.1007/978-981-15-6048-4_25.

[4] A. Sivanathan *et al.*, "Classifying IoT devices in smart environments using network traffic characteristics," *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1745–1759, Aug. 2019, doi: 10.1109/TMC.2018.2866249.

[5] N. Ammar, L. Noirie, and S. Tixeuil, " Improved identification of the type of connected objects by supervised classification," (In French), *CORES2019-Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication.*, pp. 1–5, 2019.

[6] Y. Meidan *et al.*, "ProfilIoT: a machine learning approach for IoT device identification based on network traffic analysis," in *Proceedings of the Symposium on Applied Computing*, Apr. 2017, pp. 506–509. doi: 10.1145/3019612.3019878.

[7] M. Snehi and A. Bhandari, "A novel distributed stack ensembled meta-learning-based optimized classification framework for real-time prolific IoT traffic streams," *Arabian Journal for Science and Engineering*, vol. 47, no. 8, pp. 9907–9930, Aug. 2022, doi: 10.1007/s13369-021-06472-z.

[8] B. Bezawada, M. Bachani, J. Peterson, H. Shirazi, I. Ray, and I. Ray, "Behavioral fingerprinting of IoT devices," in *Proceedings of the 2018 Workshop on Attacks and Solutions in Hardware Security*, Jan. 2018, pp. 41–50. doi: 10.1145/3266444.3266452.

[9] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A.-R. Sadeghi, and S. Tarkoma, "IoT SENTINEL: Automated device-type identification for security enforcement in IoT," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, Jun. 2017, pp. 2177–2184. doi: 10.1109/ICDCS.2017.283.

[10] M. Sneh and A. Bhandari, "Empirical investigation of IoT traffic in smart environments: characteristics, research gaps and recommendations," in *2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART)*, Dec. 2021, pp. 176–181. doi: 10.1109/SMART52563.2021.9676298.

[11] S. Naik and V. Maral, "Cyber security — IoT," in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, May 2017, pp. 764–767. doi: 10.1109/RTEICT.2017.8256700.

[12] L. Atzori, A. Iera, and G. Morabito, "Understanding the internet of things: definition, potentials, and societal role of a fast evolving paradigm," *Ad Hoc Networks*, vol. 56, pp. 122–140, Mar. 2017, doi: 10.1016/j.adhoc.2016.12.004.

[13] P. Krishnan, K. Jain, K. Achuthan, and R. Buyya, "Software-defined security-by-contract for blockchain-enabled MUD-aware industrial IoT edge networks," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 10, pp. 7068–7076, Oct. 2022, doi: 10.1109/TII.2021.3084341.

[14] "Wireshark." https://www.wireshark.org (accessed May 17, 2021).

[15] A. Sivanathan, H. H. Gharakheili, and V. Sivaraman, "Can we classify an IoT device using TCP port scan?," in *2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*, Dec. 2018, pp. 1–4. doi: 10.1109/ICIAFS.2018.8913346.

[16] "Nmap." https://nmap.org/ (accessed May 17, 2021).

[17] E. Blasch *et al.*, "Machine learning/artificial intelligence for sensor data fusion–opportunities and challenges," *IEEE Aerospace and Electronic Systems Magazine*, vol. 36, no. 7, pp. 80–93, Jul. 2021, doi: 10.1109/MAES.2020.3049030.

[18] M. Aria, C. Cuccurullo, and A. Gnasso, "A comparison among interpretative proposals for Random Forests," *Machine Learning with Applications*, vol. 6, Dec. 2021, doi: 10.1016/j.mlwa.2021.100094.

[19] B. Mahesh, "Machine learning algorithms -A review," *International Journal of Science and Research (IJSR)*, vol. 9, no. 1, pp. 381–386, 2020, doi: 10.21275/ART20203995.

[20] U. Barman and R. D. Choudhury, "Soil texture classification using multi class support vector machine," *Information Processing in Agriculture*, vol. 7, no. 2, pp. 318–332, Jun. 2020, doi: 10.1016/j.inpa.2019.08.001.

[21] M. Waleed, T.-W. Um, T. Kamal, and S. M. Usman, "Classification of agriculture farm machinery using machine learning and internet of things," *Symmetry*, vol. 13, no. 3, Mar. 2021, doi: 10.3390/sym13030403.

[22] G. Lebrun, C. Charrier, O. Lezoray, and H. Cardot, "Construction of efficient and low-complexity decision functions with SVMs," (In French), in *RJCIA*, 2005, pp. 1–14.

[23] Q. Dai, C. Zhang, and H. Wu, "Research of decision tree classification algorithm in data mining," *International Journal of Database Theory and Application*, vol. 9, no. 5, pp. 1–8, May 2016, doi: 10.14257/ijdta.2016.9.5.01.

[24] N. Ben Amor, S. Benferhat, and Z. Elouedi, "Naive Bayesian networks and decision trees in intrusion detection systems,"(In French), *TSI-Technique et Science Informatiques*, vol. 25, no. 2, pp. 167–196, 2006.

[25] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, "Random forests and decision trees," *IJCSI International Journal of Computer Science Issues*, vol. 9, no. 5, pp. 272–278, 2012.

[26] A. O. Ok, O. Akar, and O. Gungor, "Evaluation of random forest method for agricultural crop classification," *European Journal of Remote Sensing*, vol. 45, no. 1, pp. 421–432, Jan. 2012, doi: 10.5721/EuJRS20124535.

[27] D. Zhao *et al.*, "Using random forest for the risk assessment of coal-floor water inrush in Panjiayao Coal Mine, northern China," *Hydrogeology Journal*, vol. 26, no. 7, pp. 2327–2340, Nov. 2018, doi: 10.1007/s10040-018-1767-5.

[28] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "KNN model-based approach in classification," in *OTM 2003: On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, 2003, pp. 986–996. doi: 10.1007/978-3-540-39964-3_62.

[29] H. A. Abu Alfeilat *et al.*, "Effects of distance measure choice on K-nearest neighbor classifier performance: A review," *Big Data*, vol. 7, no. 4, pp. 221–248, Dec. 2019, doi: 10.1089/big.2018.0175.

[30] R. Mirtorabi, "Automating water capital activities using Naïve Bayes classifier with supervised learning algorithm," University of Waterloo, 2021.

[31] G. Gültekin and O. Bayat, "A Naïve Bayes prediction model on location-based recommendation by integrating multi-dimensional contextual information," *Multimedia Tools and Applications*, vol. 81, no. 5, pp. 6957–6978, Feb. 2022, doi: 10.1007/s11042-021-11676-4.

[32] A. Salvail-Berard, "Les arbres de décision hybrides," *Cahier de Mathématique de l'Université de Sherbrooke*, vol. 2, pp. 34–58, 2012.

[33] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, pp. 116–123.

[34] B. Gaye, D. Zhang, and A. Wulamu, "Sentiment classification for employees reviews using regression vector- stochastic gradient descent classifier (RV-SGDC)," *PeerJ Computer Science*, vol. 7, Sep. 2021, doi: 10.7717/peerj-cs.712.

[35] A. Sivanathan *et al.*, "Characterizing and classifying IoT traffic in smart cities and campuses," in *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, May 2017, pp. 559–564. doi: 10.1109/INFCOMW.2017.8116438.
[36] O. Cheikhrouhou, M. B. Jemaa, and M. Laurent-Maknavicius, "New authentication method EAP-EHash," (In French), *CFIP 2006/Francophone Conference on Protocol Engineering*. Hermes, 2006.
[37] A. Latifah, S. H. Supangkat, and A. Ramelan, "Smart building: A literature review," in *2020 International Conference on ICT for Smart Society (ICISS)*, Nov. 2020, pp. 1–6. doi: 10.1109/ICISS50791.2020.9307552.
[38] S. J. Rashid, A. M. Alkababji, and A. M. Khidhir, "Communication and network technologies of IoT in smart building: A survey," *NTU Journal of Engineering and Technology*, vol. 1, no. 1, pp. 1–18, 2021.
[39] R. Sangeetha and B. Kalpana, "Identifying efficient kernel function in multiclass support vector machines," *International Journal of Computer Applications*, vol. 28, no. 8, pp. 18–23, 2011.
[40] "Jupyter notebook." https://jupyter.org/ (accessed Oct. 20, 2021).
[41] N. Ammar, L. Noirie, and S. Tixeuil, "Autonomous identification of IoT device types based on a supervised classification," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, Jun. 2020, pp. 1–6. doi: 10.1109/ICC40277.2020.9148821.

BIOGRAPHIES OF AUTHORS

Loubna Elhaloui received a specialized master's degree in computer networks and systems from the Faculty of Sciences Ben M'sik, Hassan II University of Casablanca, Morocco. She is a teacher and researcher in computer networks at the EMSI Rabat, Morocco. she is a member of the laboratory of information technologies and modeling. She can be contacted at l.elhaloui@gmail.com.

Sanaa El Filali is currently a full professor of computer science in the Department of Mathematics and Computer Science at Faculty of Science Ben M'Sik, Hassan II University of Casablanca. She received her Ph.D. in computer science from the Faculty of Science Ben M'sik in 2006. Her research interests include computer training, the Internet of things, and information processing. She can be contacted at elfilalis@gmail.com.

El Habib Benlahmer is currently a full professor of computer science in the Department of Mathematics and Computer Science at Faculty of Science Ben M'Sik, Hassan II University of Casablanca since 2008. He received his Ph.D. in computer science from ENSIAS in 2007. His research interests span both web semantic, NLP, mobile platforms, and data science. He can be contacted at h.benlahmer@gmail.com.

Mohamed Tabaa received a degree of engineer in telecommunication and networking from the Moroccan school of engineering science of Casablanca, Morocco in 2011. He received a master's in radiocommunication, and embedded electronic systems from University of Paul Verlaine of Metz, France. He received his Ph.D. and H.D.R. diploma in electronics systems from University of Lorraine Metz, France in 2014 and 2020 respectively. Since 2015, he has been the Director of the LPRI private Laboratory attached to the EMSI. His research interests include an array of digital signal processing for wireless communications, IoT, digitalization, renewable energy, and embedded systems. He has served on the organizing committees and technical program committees of several international conferences, including IEEE International Conference on Microelectronics ICM, Innovation and New Trends in Information Systems INTIS, IEEE Renewable Energies, Power Systems and Green Inclusive Economy REPS & GIE, IEEE International Conference on Control and Fault-Tolerant Systems SysToL. He can be contacted at m.tabaa@emsi.ma.

Youness Tace holds a bachelor's degree in mathematics and computer science, a master's in big data & data science (DSBD). He is a doctoral student in science and technology and has acquired several certificates and professional skills. He currently teaches at the Moroccan School of Engineering Sciences (EMSI) and did a few visits to Ben M'Sik Faculty of Sciences to supervise and teach master's students. He is a member of the Center for Innovation and Technology Transfer (CITT). He has a penchant for the fields of the Internet of things, artificial intelligence, and web development. He can be contacted at email: youness.tace.pro@gmail.com.

Nouha Rida got her Ph.D. degree in computer science from the University Mohamed V of Rabat- Morocco. She is a full professor in Computer Science at the EMSI Rabat, Morocco. She is a member of the smartiLab, and she is a member of a Network and Intelligent Systems Group and has many research contributions. She can be contacted at email: nou-harida@gmail.com.