Complex & Intelligent Systems (2023) 9:5281–5296
https://doi.org/10.1007/s40747-023-01000-y

ORIGINAL ARTICLE

Characterization of threats in IoT from an MQTT protocol-oriented dataset

Ángel Luis Muñoz Castañeda[1] · José Antonio Aveleira Mata[2] · Héctor Aláiz-Moretón[3]

Received: 28 February 2021 / Accepted: 7 February 2023 / Published online: 17 March 2023

Abstract
Nowadays, the cybersecurity of Internet of Thing (IoT) environments is a big challenge. The analysis of network traffic and the use of automated estimators built up with machine learning techniques have been useful in detecting intrusions in traditional networks. Since the IoT networks require new and particular protocols to control the communications between the different devices involved in the networks, the knowledge acquired in the study of general networks may be unuseful some times. The goal of this paper is twofold. On the one hand, we aim to obtain a consistent dataset of the network traffic of an IoT system based on the Message Queue Telemetry Transport protocol (MQTT) and undergoing certain type of attacks. On the other hand, we want to characterize each of these attacks in terms of the minimum possible number of significant variables allowed by this protocol. Obtaining the data set has been achieved by studying the MQTT protocol in depth, while its characterization has been addressed through a hybrid (filter/wrapper) feature selection algorithm based on the idea behind the minimum-redundancy maximum-relevance (mRMR) algorithm. The dataset, together with the feature selection algorithm, carries out a characterization of the different attacks which is optimal in terms of the accuracy of the machine learning models trained on it as well as in terms of the capability of explaining their underlying nature. This confirms the consistency of the dataset.

Keywords IoT · MQTT · Machine learning · Features selection

Introduction

The number of home devices that have sensors connected to the Internet or a local network has increased exponentially. These devices are designed to program events, collect information, or offer remote control capacities, retrieving a large amount and variety of data. The interconnection of these devices is what we call the Internet of Things (IoT).

✉ Ángel Luis Muñoz Castañeda
amunc@unileon.es

José Antonio Aveleira Mata
javem@unileon.es

Héctor Aláiz-Moretón
hector.moreton@unileon.es

[1] Department of Mathematics, Universidad de León, León, Spain

[2] RIASC, Universidad de León, León, Spain

[3] Department of Electrical Engineering and Systems and Automatic, Universidad de León, León, Spain

According to Cisco [16], the number of connected devices is expected to reach 500 billion by 2030, making security in IoT systems of paramount importance. Security in IoT systems has special characteristics due to the wide variety of different networks and protocols used in them [18]. The devices often have limited processing capacity to be efficient, and security in IoT devices is often ignored or treated as a late occurrence. Furthermore, the short time to market and reduced costs that drive the design and development of the device do not help to alleviate this problem [64]. Because of this, different types of known vulnerabilities can be found in commercial baby monitoring devices (see [56], for instance).

Attacks on IoT systems can affect their functionality, with alterations in system messages, alterations in the operation of actuators, or the theft of private information gathered by the sensors. The security of IoT devices can also affect the general security of the Internet. For example, in 2012, Carna botnet [49] revealed that there were more than 1.2 million open devices that allowed logins with empty or default credentials. In January 2014, an Internet-connected fridge was discovered as a part of a botnet sending over 750,000 spam

e-mails [57]. One of the most significant recent attacks, the Mirai attack in September 2016 [36], used the special features and vulnerabilities of IoT devices, their low level of security, and the large number of them that are always connected to the Internet, to infect them and create a botnet which attacked the service provider, Dyn, with a distributed denial-of-service attack (DoS). This attack took down hundreds of websites, including Twitter, Netflix, Reddit, and GitHub for several hours.

The analysis of network traffic has been useful in detecting intrusions in traditional networks thanks to the use of intrusion detection systems (IDS). These systems are improved by adding new detection rules or automatic classifiers obtained using Machine Learning (ML) techniques, and are tested with datasets that contain network traffic with normal traffic and traffic under attack. On the other hand, the strategies to improve an IDS mentioned above require prior knowledge about how the attacks we want to detect perturb the network in terms of the variables allowed by the protocol that governs that network. That is to say, we need to characterize the attacks inside the features space allowed by the governing protocol. Such characterization is usually faced making use of feature selection algorithms. However, such algorithms need to be fed with a dataset that contains network traffic of both types, normal and with traces corresponding to situations under attack. From everything said so far, it follows that these types of datasets are of great importance [41] and the task of capturing the information generated in an IoT network becomes a critical step in analyzing intrusions in the IoT [67]. Nevertheless, there are almost no public datasets with network information on security-oriented IoT systems that capture attacks on the particular vulnerabilities of these systems. Most datasets gathered from IoT attacks contain information from sensors that are oriented to optimize the system or learn about trends (see, for instance, [2,12,37]).

The goal of this article is twofold:

1. On one hand, to construct a consistent dataset of the network traffic of an IoT system based on the MQTT protocol and undergoing three types of attacks: Man-in-the-middle (MITM), Denial of Service (DoS), and Intrusion (I).
2. On the other hand, to determine, for each type of attack, a set of features among those allowed by the MQTT protocol that let us to distinguish the attack from normal traffic accurately and to explain the underlying nature of the attack. We require for the features to have the least correlation between themselves as possible to avoid redundancies, and to be as few as possible to avoid unnecessary complexity.

To obtain a dataset of the network traffic of an IoT system undergoing an attack, the protocol entitled Message Queue Telemetry Transport (MQTT), widely used in IoT [3], has been studied in depth. When its working has been verified and its vulnerabilities studied, three types of attack have been carried out in an environment that simulates a real situation to capture network traffic undergoing these attacks and to investigate whether is it possible or not to characterize these attacks in terms of the trace left by them on the network.

To address the characterization problem subject to the conditions imposed, the most suitable procedure seems to be the minimum-redundancy maximum-relevance (mRMR) algorithm [22]. However, this algorithm has two drawbacks. First, it requires the user to fix the number of features that the algorithm will output. Second, it does not take into account any predictive model which let us to measure the goodness of the selected features for a prediction task. Therefore, the characterization has been addressed through a hybrid (filter/wrapper) feature selection algorithm based on the idea behind the mRMR feature selection algorithm. The score function used for the filter process is the normalized mutual information function (NMIF), while the wrapper process is carried out using a model selection algorithm in which the models are trained and tested over the reduced subsets of features provided by the filter process. The algorithm provides, for each type of attack, a small subset of features with low correlation between themselves and high correlation with the class variable. As a by-product, we get classification models for the different attacks. The generated predictors show a high accuracy in the testing stage, so the selected features give a good characterization of the attacks considered in this work.

The advantage of using an information-theoretic measure as the NMIF function is that the features provided by the algorithm explain the underlying nature of the IoT attacks considered in this work by themselves. Therefore, the algorithm may be useful to get expert knowledge by applying it to less-known types of attacks. Also, it may be useful to define detection rules to improve an IDS.

The dataset, together with the feature selection algorithm, then performs a characterization to obtain the most relevant variables to detect the different attacks, which confirms the consistency of the dataset.

Related work

As regards IoT security, there are two different lines of research. On the one hand, we have those problems related to the security of the whole network and, on the other hand, those related to the security of an isolated device in the network. The problem that concerns us in this work is a mixture of both. More specifically, we ask whether is it possible or not to distinguish the kind of an attack that is taking place on a single device in an IoT network by enabling machine learning models trained with the network traffic data.

In general, datasets that contain network information on systems under attack are very useful for feed intrusion detection systems (IDS) [42]. The biggest trouble is finding IoT network dataset composed by regular and anomalous traffic.

A well-known dataset is KDDD99 [34] which gathers network traffic over the TCP protocol in a system in which different attacks, such as DoS, User to Root (U2R), Remote to Local (R2L) and Probing Attack, are made and tagged. With this dataset, studies have been conducted for the development of IDS for IoT environments that focus on denial of service attacks performed by botnets of infected IoT devices [45,50]. Although KDD99 is still a valid dataset, it is not sufficient for more modern networks with new protocols for IoT environments [55], being useful datasets like the one proposed for IoT environments.

Another dataset to highlight is the AWID dataset [35], which gathers TCP frames of data from a WLAN network over which several attacks have been made on the 802.11 security mechanism (i.e. WEP, WPA, WPA2) such as ARP Injection or Dictionary Attack. They use machine learning techniques to detect attacks on Wi-Fi networks with work that focuses on IoT, such as Impersonation attacks [7,51].

When using the dataset to detect anomalies, it is possible that not all of the features considered give relevant information, and that some of them contain false correlations that make detection difficult, which leads to a decrease in accuracy and an increase in computational complexity. Feature selection algorithms and dimensional reduction are used for the characterization of datasets with statistical methods, information theory and machine learning techniques to optimize IDS and reduce the IDS model's complexity.

Regarding the KDD99 dataset, in [14] the authors perform a characterization of the dataset to detect some attacks to IoT systems showing, as a result, that the characterization process improves the accuracy of IDS systems. However, they are for more generic attacks on IoT systems. Many other studies have been carried out on the reduction of its features to obtain the most relevant ones for the detection of attacks, see [27] for instance. For the AWID dataset, in [62] the authors make a selection of optimal features with a classifier that uses Support Vector Machines (SVM) and redundant features are removed using PSO-based algorithms. In [4], the authors implement Deep-Feature Extraction and Selection (D-FES), which combines stacked feature extraction and weighted feature selection on the AWID benchmark data set. In [39], the authors deal with the unbalance issue in the AWID dataset. They perform a characterization of anomalous traffic that avoids this unbalance using algorithms such as Word2Vec, KMeans and SMOTE.

In [33], the authors provide a survey on the features selection and machine learning algorithms used to face the intrusion detection problem in traditional networks in the last years. Regarding deterministic algorithms for features selection, the authors show that most of the research in the area is supported on algorithms based on Mutual Information, Entropy, Correlation Coefficient, Chi-Square, Relief-F, and Gain Ratio. Furthermore, they show that Mutual Information and Gain Ratio based algorithms are by no means the most used methods for selecting features in Intrusion Detection problems on general networks.

Regarding the special case of IoT, there are significant works on security that have been carried out based on the analysis of network traffic using machine learning techniques.

In [23], for example, the authors analyze how machine learning can help in detecting the activity of an IMSI catcher in a mobile network. It puts the devices that are in the neighbourhood under a man-in-the-middle (MitM) attack by trying to be the preferred base station in terms of signal strength. In [26], the authors propose an anomaly detection scheme with feature selection using the boruta algorithm based on Random Forest (RF) classification technique with good results using for the detection of intrusions with the DARPA [1] dataset with network traffic. On the other hand, [46] propose an intrusion detection and mitigation framework (IoT-IDM) for the protection of a network of intelligent devices implemented in domestic environments. IoT-IDM supervises the network activities of the devices and investigates if there is any irregular activity. When an intrusion is detected, IoT-IDMt is able to block the intruder. In [60], the authors use a convolutional neural network model to create a multiclass classification model. The proposed model is then implemented using convolutional neural networks in one, two, and three dimensions. In the pre-processing phase, the authors use a model-based feature selection technique called RFE (Recursive Feature Elimination) to select 64 relevant features. In [61], a framework for detecting anomalies in IoT networks is described. Using conditional Generative Adversial Networks (GANs), the authors generate real-world distributions for a given feature set to face the problem of data imbalance. The performance of the GAN models in classification tasks are evaluated using a Feed Forward Neural Network and tested on two network-based anomaly detection datasets and five IoT network-based anomaly detection datasets. The authors do not filter the set of features provided by the protocol. In [43], the authors propose a Deep Learning (DL) based Network IDS trained using a public dataset containing MQTT attacks. Again, the authors do not filter the set of features provided by the protocol. In [63], the authors create a dataset with network traffic in a IoT system governed by the MQTT protocol and under certain types of attacks. They validate the dataset by training and testing some common machine learning models. In the preprocessing phase, the authors filter the full set of features. However, the process is not done automatically but by hand attending at certain criteria. The filter process ends up with 33

selected features and they do not specialize the process to the types of attacks. In [31], the authors evaluate the effectiveness of several machine learning models to detect MQTT-based attacks. The authors consider three abstraction of features, namely, packet-based (29 features), unidirectional flow (18 features), and bidirectional flow (18 features). The authors dropped certain features to avoid specific features influence before giving the list of used features. Apart from this, there is no feature selection process. The authors train and test machine learning models using the three type of features separately showing the importance of bidirectional features in the classification task. In [15], the authors described the implementation of a lightweight anomaly-based IDS for IoT networks, focusing on attacks on MQTT. They build a dataset that contains attacks to the network, and then train and test several common machine learning models on it. Regarding the pre-processing phase, the authors make use of a model-based feature selection algorithm, the SelectKBest method, to select 24 relevant features.

In Table 1, we give a summary of the datasets considered in the works mentioned above, used to investigate cybersecurity in IoT environments.

Despite the importance of entropy based methods (such as mutual information and information gain) for selecting features for anomalies detection in traditional networks [33], the analogous problem when dealing with IoT environments has not been faced with such methods yet. It is especially surprising taking into account the advantages of such methods, which have been proved theoretically and in practice. Furthermore, even the appearance of datasets with IoT abnormal traffic very recently, no of them cover the type of attacks considered in this work. This motivates the construction of the dataset.

Although the existing gap regarding the use of Mutual Information based algorithms for selecting and characterizing attacks taking place in IoT networks is enough to motivate this research, there exists a theoretical motivation, as said above, which is worth pointing out. In [11], the authors prove that to maximize the conditional likelihood of the training labels (under certain filter assumption) in a given machine learning problem (no matter the context of the problem), is equivalent to find the minimum set of features that minimizes the conditional Mutual Information $I(X_{\overline{\theta}}; Y|X_\theta)$ (see [11, Sect. 3.2]). Here, θ is a binary vector of the same length as the available set of features where the ith component is equal 1 if the corresponding feature is taken into account, and 0 otherwise. Furthermore, they prove that IAMB algorithm [59] is in fact a greedy iterative maximization of the conditional likelihood [11, Corollary 6], and that the Joint Mutual Information (JMI) criterion [59] provides the best tradeoff in terms of accuracy, stability, and flexibility with small data samples. Note that minimizing $I(X_{\overline{\theta}}; Y|X_\theta)$ means, in certain sense, to find the most meaningful features avoiding features that are redundant, which is the basis of mRMR algorithm [21].

In this article, we give a feature selection algorithm based on the ideas behind mRMR algorithm, close to that of [32], in which the goal is to find a balance between the maximization of the accuracy of a (ensemble of) machine learning models and the maximization of the conditional likelihood of the training labels.

Procedure for IoT dataset compilation

Due to the lack of IoT network environment datasets, we have designed and implemented a procedure for getting this kind of well-structured data. This section is devoted to describing this procedure. A real environment is created for data collection. The environment scale allows to collect all the traffic generated by the IoT environment Local Area Network (LAN). The environment also collects generic network traffic when for example is browsing the Internet, obtaining all the real traffic generated in the network, in not only the specific traffic of the MQTT protocol.

Message Queue Telemetry Protocol (MQTT)

The Message Queue Telemetry Protocol (MQTT) is an IoT protocol widely used in IoT systems, because of its performance [30]. The MQTT protocol is a light publication/subscription messaging protocol, which works on TCP, designed for M2M (machine to machine) communications and very useful for connecting devices in networks with low bandwidth [29]. Its architecture follows a star topology with a central node that acts as a server or broker, which is responsible for managing the network and transmitting messages.

The communication is based on topics. A client of the broker publishes the message on a topic and the clients that wish to receive it must subscribe to this topic. The communication can be 1 to 1, or 1 to N and real time. A topic is represented by a string with a hierarchical structure. Each hierarchy is separated by a '/'. The operating architecture of the MQTT protocol can be seen in Fig. 1.

A test environment consisting of several IoT devices and a set of web applications that interacts with them has been designed and developed for simulating a real IoT system that uses the MQTT protocol and performs several attacks to gather the generated traffic with a packet-based data collection [68]. The test implemented runs on a LAN in the following way:

(a) We use a server that hosts the web application and serves as a broker of the MQTT protocol. Node.js is used to develop this server because of its efficiency in controlling many simultaneous connections with respect to other

Table 1 Summary of available IoT datasets that contain anomalous network traffic

Dataset	Traffic type	IoT attacks
KDD99	Not focused on IoT context	Generic DoS attacks
AWID	Not focused on IoT context	DDoS and Impersonation
IoT-23	Focused on DNS traffic for IoT context	Botnets
MQTT-IoT-IDS2020	Simulated environment with only MQTT traffic	Scanning and brute-force
MQTT-set	Simulated environment with IoT-Flock tool only MQTT traffic	Brute-force, malformed data and DoS
Bot IoT	MQTT protocol (weather station) and AWS network traffic	Attacks by generic bots that can affect IoT, such as Dos and DDoS
Aretmis	Network with MQTTc	Only DoS attacks on the MQTT network
Proposed MQTT	All traffic in the environment and the IoT protocol	MQTT specifics vulnerabilities attacks

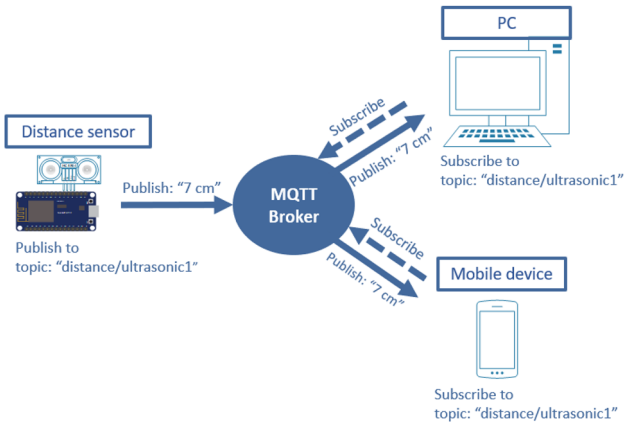


Fig. 1 MQTT publish/subscribe architecture

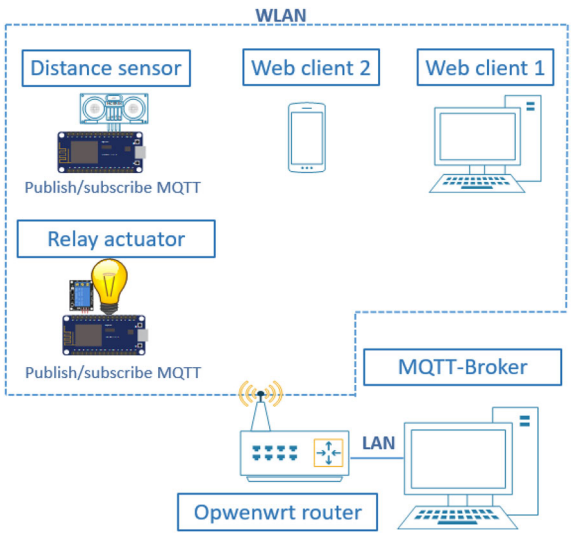


Fig. 2 Development of the testing environment schema

server technologies [13], so that this server works as a Broker as used by the "Mosca" library [17].

(b) To interact with the different sensors and actuators, an application is required that offers an interface. In this case it was decided to develop an application with the modern web angular.js technology, which connects to the broker as another client with the Angular-MQTT library [40], instead of using the MQTT protocol for communication with the broker, it uses WebSocket.

(c) An integrated board with a WiFi connection is required to connect the various sensors and actuators, the NodeMCU boards are selected, which is a low cost micro-controller that has a WiFi connection through the ESP 8266 chip. Furthermore, these boards have GPIO inputs that connect with the different devices (actuators and sensors), The micro-controllers were programmed in C ++ using the Arduino IDE and we used the PubSubClient library to communicate with the broker using the MQTT protocol as clients.

Two NodeMCU modules are used as follows:

- Connection of an ultrasonic sensor HC-SR04, which signs up to the topic "distance/ultrasonic1", this end-point publishes the distance of any element located in front.
- Connection with an actuator that consists of a switch that turns the light of a lamp on and off, this device subscribes to the topic "light/relay" and depending on the numerical value it changes its status: "0" Off "1" On.

Smartphones and PCs generate MQTT traffic at the time of connecting, using the Wi-Fi connection and interacting with the devices in the environment Fig. 2. They will also generate a new normal traffic because of the Internet connection with the purpose of simulating a real system.

Attacks recompilation

Different attacks on the MQTT protocol are carried out on the previously described environment. The environment captures the traffic generated by the attacks as well as all the traffic in the designed environment. The attacks have been chosen taking into account the particular functionality of the MQTT protocol and the specific vulnerabilities of the specification [47], with attacks on the broker such as the DoS attack, attacks on the implementation of the protocol such as the use of the well-known port as the intrusion attack, and the alteration of the messages as MQTT is a lightweight protocol with the MiTM attack.

To begin with, we will describe which are the actions implemented to simulate each type of attack and how this actions interact with the network taking into account the IoT protocol (MQTT) rules. With this in hand we will able to check whether the selected features describe the underlying nature of each type of attack.

The denial of service attack

The denial of service attack (DoS) is one of the most common attacks on the Internet [52]. This attack forces the system to refuse routing messages or redirects these messages to where they should not go. This fact is one of the challenges of cybersecurity in IoT. In the MQTT protocol, the broker manages all the connections and could be the victim of a DoS attack causing a malfunction in the whole system because the clients of the broker do not receive the correct messages [6].

A simulation of a DoS attacks has been carried out in a test environment using the MQTT-malaria program [48], this program is used for testing the scalability and load testing utilities for MQTT environments. With the MQTT-malaria program command "malaria publish" it can imitate several clients separately by publishing messages of a specific size and indicate the speed of messages per time period. With this tool we generate a great amount MQTT traffic so the broker has to manage a large number of requests. As a result, the server will be locked for a period of time. Several attacks from a computer in the test environment have been made. All the traffic generated by the normal working of the system is gathered with Internet browsing interaction with the devices, the video download and the traffic generated by the DoS attacks.

Man in the middle attack

The man-in-the-middle (MitM) attack occurs when an attacker is able to observe and intercept messages and modify them without either of the two end communications points knowing that the information has been modified.

An MitM attack between the one device, in this case the sonar, and the server, with the objective of modifying the MQTT packets sent by the sonar to the server has been made. A Kali Linux distribution with the Ettercap tool has been chosen for carrying out this attack by scanning the hosts of the network and adding the addresses in which the attacker is placed in the middle.

The next step is to modify the MQTT messages that the sonar sends to the server. To do so, the nfqsed [28] program modifies network traffic using a predefined set of substitution rules that modifies the values sent by the sonar to the server.

Intrusion by other MQTT clients

One of the specific vulnerabilities of using the MQTT protocol arises when there is no authentication to access the broker and therefore, by scanning the well-known MQTT port 1883, it is possible to know which servers use this protocol and which are available. For example, with the Shodam scanner, it is possible to get a large number of unprotected brokers. Once the server is detected it is possible to see which topics are being managed by the broker using the special character "#" [5], This can be used by an external attacker to find out the active topics available for subscribing to. Therefore, the attacker could gather significant information or publish false information on them.

The intrusion attack is carried out from a MQTT Mosquitto [44] client, that subscribes to the "#" topic so that all the information generated by the other clients as well as the topics to be used are obtained.

With the information obtained by listing all topics, the attacker can connect with the same Mosquitto client to publish false information on both the sensor and the relay. It is also necessary to gather the normal traffic generated to complete the dataset.

Creation of the datasets

As previously mentioned, the data is generated by the WiFi network from a router with Openwrt using the tcpdump command to gather traffic in PCAP files.

To carry out a traffic analysis, it is necessary to separate the information in the PCAP file into the relevant traffic for each protocol. Because of the complexity of PCAP files, a custom application entitled "Web Dissector"[1] dissects files into these fields and offers the versatility of changing the fields quickly and easily according to the needs of the research. With this tool it loads a PCAP file and indicates the fields to be dissected and returns a CSV file with the attacks tagged. Thus datasets of great use in feeding supervised and unsupervised

[1] Web dissector is a tool developed by SECOMUCI research group of University of León (Spain).

machine learning techniques are obtained. ML (machine learning) algorithms [58] are feed with the dataset with the aim of generating intrusion detection models. The study of all the datasets can also reveal valuable information on the conditions under which these intrusions take place and to predict intrusions.

Materials and methods

This section details the different techniques used for accomplishing the objectives of this research, together with the datasets extracted that will be characterized.

IoT-MQTT dataset

The datasets obtained contain 67 fields as described below:

- 28 Fields common to all the frames of the gathered traffic are selected and offer relevant information in all cases. Among these fields are the system times, the relative time of collating, the origin and destination of the MAC and IP addresses and fields at frame level. These fields are taken to detect patterns in frame attacks that are not directly related to the specific IoT protocol used.
- We have obtained from Wireshark Display Filter Reference the 38 fields that compose the specification of the MQTT protocol. This allows an in-depth analysis of what is happening in the IoT system when this protocol is used.
- A "type" field that labels the frames under attack with the name of the attack "DoS" "MitM" and "intrusion", and the remaining the frames are labelled "normal".

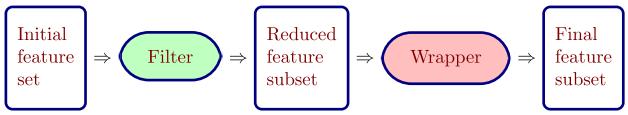
The three files generated from the dataset correspond to each of the attacks carried out on the system on the MQTT protocol which are as follows:

- DoS.csv that contains the collation of 94,625 frames of which 45,513 are under attack traffic and 49,112 are normal traffic, in this case there are many under attack frames because this type of attack is used to generate a lot of traffic.
- MitM.csv contains 110,668 frames with 3,855 under the man-in-the-middle attack and 106,813 under normal traffic frames.
- Intrusion.csv with 80,893 total frames with 1,898 are frames produced by the intrusions of clients outside the system and 78,995 normal traffic frames.

Feature selection algorithm

Feature selection algorithms attempt to find the set of variables that allow machine learning models to be trained with the best possible performance and is a crucial step in the general machine learning schema. Among others, there is a step in any feature selection algorithm that determines its nature (see [54]), and regards the strategy used to evaluate subsets of features within the initial set of features. There are mainly two kinds of methods for dealing with this strategy: wrapper methods and filter methods.

Wrapper methods for selecting features make use of predictive models to solve the problem. Roughly speaking, they train and test a predictive model with each subset of features and the solution is given by the subset of features with which the predictive model performs better. On the other hand, filter methods for selecting features make use of scoring functions that do not depend on predictive models but on the relationships between the initial characteristics of the data and a target variable. Thus, they can be considered as a pre-processing step in the general machine learning scheme. Although these two types of methods are slightly different, they can be combined to obtain hybrid feature selection algorithms. The general work flow of these hybrid algorithms is as follows:



In this work, we propose a hybrid feature selection algorithm based on the mutual information function, and our goal is two-fold. On the one hand, we aim to find a predictive model with good performance. And, on the other hand, we attempt to find a minimum set of significant features that explains the underlying nature of the concept we want to learn, i.e., that characterize an IoT threat. Attending to this requirement, we have chosen a theoretic information measure (normalized mutual information function) as a the scoring function of the filter process. The proposed algorithm is a hybrid algorithm, so it is made up of two routines; that corresponding to the filter process and that corresponding to the wrapper process.

The filter process

The idea behind the proposed filter process is based on the minimum-redundancy maximum-relevance (mRMR) feature selection algorithm [22]. We denote by $\mathrm{MIF}(X, Y)$ the mutual information function of two random variables. This is a non negative function. Likewise, we denote by $\mathrm{NMIF}(X, Y)$ the normalized mutual information function of two random variables. In this case, $\mathrm{NMIF}(X, Y) \in [0, 1]$ (see [38,65] for the original definition of normalized infor-

mation function and for a comparison with other information measures). In both cases the value 0 is obtained when both variables are independent, while the value 1 is obtained in the second case when there is a perfect correlation.

The mRMR algorithm solves the optimization problem given by

$$\min_S \left\{ \frac{1}{|S|^2} \sum_{i,j \in S} \mathrm{MIF}(X_i, X_j) \right\},$$
$$\max_S \left\{ \frac{1}{|S|} \sum_{i \in S} \mathrm{MIF}(X_i, type) \right\},$$

where the subset S of the set of features is assumed to be the same in both equations. Observe that this method can discard significant features that contain information not contained in other features. To avoid this we propose a slightly different method by considering the normalized mutual information function.

Given a real number $\alpha, \beta \in [0, 1]$, a variable X is called α-significant with respect to another variable Y if NMIF(X,Y) $\geq \alpha$, while a variable X is called β-redundant with respect to a set of random variables S if NMIF(X,Y) $\geq \beta$ for some $Y \in S$. Given α, β as before, we will filter the set of features to get a subset formed by α-significant variables not containing β-redundant variables.

FilterRoutine:
Input: $\alpha, \beta \in [0, 1]$.

1. Compute $\alpha_i := \mathrm{MIF}(X_i, type)$ for every variable, $type$ being our target variable.
2. Let $\mathrm{Imp}(\alpha) := \{X_i \,|\, \alpha_i \geq \alpha\}$ be the subset of α-significant variables. A variable X_i is then said to be lower than X_j if $\alpha_i \leq \alpha_j$. Then we sort $\mathrm{Imp}(\alpha)$ from lowest to highest.
3. Compute the mutual information function score $\alpha_{i,j} := \mathrm{MIF}(X_i, X_j)$ for every pair of variables $X_i, X_j \in \mathrm{Imp}(\alpha)$ with $i \neq j$.
4. Let $X_0 \in \mathrm{Imp}(\alpha)$ be the lowest variable. If X_0 is β-*redundant* with respect to $\mathrm{Imp}(\alpha)$, we remove X_0 from $\mathrm{Imp}(\alpha)$, otherwise we keep it. We repeat this action with $X_1 \in \mathrm{Imp}(\alpha) \setminus \{X_0\}$, and so on. Eventually, we get a subset $\mathrm{Imp}(\alpha, \beta) \subset \mathrm{Imp}(\alpha)$ of variables.

Output: $\mathrm{Imp}(\alpha, \beta)$.

The steps described above depend on the two real numbers given as input, α and β. So varying these two parameters we arrive at different solutions, $\mathrm{Imp}(\alpha, \beta)$. This yields a map

$$\textbf{FilterRoutine} : [0, 1] \times [0, 1] \longrightarrow \left\{ \begin{array}{l} \text{subsets of the} \\ \text{initial set} \\ \text{of features} \end{array} \right\} \quad (1)$$
$$(\alpha, \beta) \mapsto \mathrm{Imp}(\alpha, \beta)$$

The wrapper process

The wrapper process consists of a selection model algorithm and the output is given by the accuracy on the test data of the best model.

WrapperRoutine:
Input: $M_1(\gamma_1^1, \ldots, \gamma_{m_1}^1), \ldots, M_n(\gamma_1^n, \ldots, \gamma_{m_n}^n)$ are machine learning models which we assume depend on certain hyper-parameters and S is a subset of features.

1. We apply hyper-parameter selection algorithms to get those values of γ_i^j that make the models to be as more accurate as possible.
2. The best model is selected and its accuracy on a test data, which we denote by $acc(S)$, is assigned to the subset of features S.

Ouput: $acc(S)$.

The steps described above yields a map:

$$\textbf{WrapperRoutine} : \left\{ \begin{array}{l} \text{subsets of the} \\ \text{initial set} \\ \text{of features} \end{array} \right\} \longrightarrow [0, 1] \quad (2)$$
$$S \mapsto acc(S)$$

The algorithm

The composition of the filter (1) and the wrapper (2) processes leads to a map:

$$\Psi : [0, 1] \times [0, 1] \longrightarrow [0, 1]$$
$$(\alpha, \beta) \mapsto acc(\mathrm{Imp}(\alpha, \beta))$$

and the problem we want to solve can be formalized as an optimization problem as follows:

$$\max_{\alpha,\beta} \{\Psi(\alpha, \beta)\}. \quad (3)$$

Observe that the function Ψ is not continuous but constant in certain squares inside $[0, 1] \times [0, 1]$. This implies that Ψ induces a partition of the square, as shown in Fig. 3, where points with the same color are points at which Ψ attach the same value. This finally leads to the proposed algorithm to solve (3). This consists of the following steps: we set initial parameters, α_0 and β_0, to define α_0-importance and β_0-redundancy, as well as step sizes μ and η to walk inside the square $[0, 1] \times [0, 1]$. At the iteration (i, j)th, the α-importance and the β-redundancy are defined by the parameters $\alpha = \alpha_0 - i \cdot \mu$ and $\beta = \beta_0 + j \cdot \eta$. The filter process gives us a subset $\mathrm{Imp}(\alpha, \beta)$ of features and the wrapper process gives us an accuracy, $acc_{i,j}$. The algorithm returns the subset $\mathrm{Imp}(\alpha, \beta)$ that maximizes the accuracy.

Algorithm 1 FeaturesSelction

Inputs: $\mu, \eta, \alpha_0, \beta_0 \in (0, 1)$. $M_1(\gamma_1^1, \ldots, \gamma_{m_1}^1), \ldots, M_n(\gamma_1^n, \ldots, \gamma_{m_n}^n)$ are machine learning models which we assume depend on certain hyper-parameters.
Output: The subset of features $\text{Imp}(\alpha_i, \beta_j)$ with higher score acc_{ij}.
Initialize $i_{max} = j_{max} = 0$
while $\alpha_i = \alpha_0 - i \cdot \mu \geq 0$ **do**
  while $\beta_i = \beta_0 + j \cdot \eta \geq 0$ **do**
    FilterRoutine with α_i, β_j as input parameters. This yields a set of variables $\text{Imp}(\alpha_i, \beta_j)$.
    WrapperRoutine with $\text{Imp}(\alpha_i, \beta_j)$ as input parameters. This gives a score $acc_{i,j} := acc(\text{Imp}(\alpha_i, \beta_j)) \in (0, 1)$.
    if $acc_{i,j} > acc_{i_{max}, j_{max}}$ **then**
      $i_{max} = i$, $j_{max} = j$
    end if
    $j = j + 1$
  end while
  $i = i + 1$
end while
return Subset of features $\text{Imp}(\alpha_{i_{max}}, \beta_{j_{max}})$

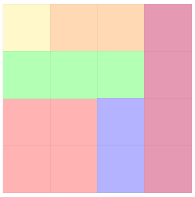


Fig. 3 Partition of the square $[0, 1] \times [0, 1]$ induced by Ψ

Experiments

To apply the feature selection algorithm described above we will have to adjust the parameters on which both the filter and the wrapper processes depend.

Set-up of the wrapper process

To run the wrapper process we consider the following state-of-the-art machine learning algorithms: Adaboost, Decision Tree, Random Forest, Gradient Boosting, Logistic Regression and SVM.

Decision trees (DT) are non-parametric supervised models. The prediction is achieved by learning simple decision rules given by the features. There are several algorithms available for training a decision tree, but the one that we will use is CART [10]. Adaboost (AdB) [24] is a meta-estimator that fits a set of weak learners, assumed to be better than random guessing models, on repeatedly modified versions of the data. The predictions given by the weak estimators are combined through a weighted sum to give the final prediction. Random forest (RF) [9] is also a meta-estimator that fits several decision trees and uses averaging to improve accuracy in the single predictions and to control overfitting. Each tree in the random forest is built from a sample drawn from the training set. When splitting a node during the learning algorithm of each tree, the split chosen is the best split from among a random subset of the features [8]. Gradient Boosting (GB) [25] is again a meta-estimator that builds models, generally decision trees, not independently as Random Forest does but in a forward stage-wise way Logistic regression (LR) is a linear model made up of the sigmoid function, which means that the target value is expected to be the sigmoid function applied on a linear combination of the input variables. The solvers we chose for our problems are liblinear, which uses a coordinate descent algorithm [66], SAG [53] and SAGA [20]. Support Vector Machines (SVM) work by constructing hyperplanes in vector spaces, where we can consider a good separation if a hyperplane has the largest distance to the nearest training data point of any class [19]. In general, the larger the margin (this distance) the lower the error of the classifier.

To search for the best hyper-parameter configuration for each of the aforementioned machine learning models, we have used Grid Search algorithm with cross validation (3 splits). A grid search is an exhaustive search through a specific subset of a hyper-parameter space of a learning algorithm. We have used the GridSearchCV method from the Scikit-learn library to carry out the hyper-parameter search. The hyper-parameters we have tuned, together with the grids, are listed in Table 2.

These hyper-parameters have been chosen because of their high influence on the respective machine learning algorithms. We give a brief description of them in Table 3.

Set-up of the filter process

The filter process depends on four parameters α, β, μ, η. We initialize $\alpha_0 = \beta_0 = 0.5$. The coefficients, μ and η, that moves α_0 backwards and β_0 onwards are fixed and both are equal to 0.1.

Results and discussion

In this section, we describe the results and reflections based on the aforementioned experiments.

The algorithm finishes after 25 iterations. The best models (with the tuned hyper-parameters) together with their accuracies and the features selected by the filter routine of the algorithm are listed in Tables 4, 5 and 6 for the datasets DoS, man-in-the-middle and intrusion respectively. The models that appear in bold are the most accurate for each data set and, therefore, the output of the algorithm applied to each data set is precisely the set of features appearing in the selected features box corresponding to these models. The summary of the outputs of the feature selection algorithm is given in Table 7.

By analyzing the features that the algorithm has selected (see Table 8 for a brief description of them), we can say how the IoT attacks are reflected in the network traffic and we can explain their underlying nature. This implies that these attacks have been characterized successfully and therefore,

Table 2 Tuned hyper-parameters and grids

Model	Max_depth	n_estimators	Loss	Learning_rate	Penalty	Solver	Kernel	C
Decision tree	5, 10, 15, 20, 50,75,100							
Random forest	5, 10, 25, 50, 100	5, 10, 25, 50, 100, 100						
Ada boost	10, 25, 50, 100, 150, 200							
Gradient boosting	5, 10, 25, 50, 75, 100	25, 50, 75, 100	dev exp	0.1, 0.01, 0.001				
Log regression					l2	liblinear sag saga		0.5, 0.75,1, 2,5,10
SVC							rbf	0.5, 1, 2, 5

Table 3 Description of hyper-parameters

Hyper-parameter	Description
Max_depth	The maximum depth of the tree
n_estimators	The number of trees in the forest
Loss	The loss function to be optimized
Learning_rate	Positive rational number by which the contribution of each tree is shrunk
Solver	Algorithm to use in the optimization problem
C	Inverse of regularization strength. Smaller values specify stronger regularization
Kernel	Specifies the kernel type to be used in the algorithm

the data set is consistent and the feature selection algorithm carries out the selection task well.

1. *Man-in-the-middle*: In an MITM attack, the attacker gets in between the device and the broker and can modify the MQTT messages sent by the device, the variables **frame.cap-len** and **mqtt.len** indicate perturbations at the network level as well as at the protocol level, **mqtt.hdrflags** anomalies in the control headers of intercepted messages. As well as irregularities in the messages sent by different customers **mqtt.msg**.
2. *DoS*: A DoS attack consists of saturating the broker by sending many connections and messages simultaneously. We can see an association between selected characteristics and the nature of the attack. The features **mqtt.duplfag** and **mqtt.qos** are relevant when new connections are produced and in DoS these are abundant. Furthermore, the size of the packages is important especially as the bigger the size of the load in every package, the faster the server will saturate. The information on the size is given mainly by **mqtt.len**, which is influenced by **mqtt.topic**.
3. *Intrusion*: During an intrusion attack, dangerous clients look at the topics and messages that the system uses to send false messages with a false client. In this case the fields are related to the new connections. For example, **mqtt.dupflag** occurs with new connections. The field **mqtt.msgtype** varies when there is a new client connection or when a client disconnects. The feature **mqtt.retain** switches when a last connected client publishes a message and the server retains it. The field **mqtt.topiclen** indicates the topics of the attackers may be different from those used regularly. This is important because in this type of attack, the attackers are incorporated by generating new connections.

After analyzing the features chosen by the feature selection algorithm, we show the existence of a strong relationship to the nature of the different attacks to the MQTT protocol taken into account in this work. This is valuable because it allows experts to know what variables should be taken into account to define rules for an IDS to be able to detect specific malicious traffic.

Table 4 Models performance and selected features for Dos

Model	Best hyper-parameters	Accuracy	Selected features
Decision tree	*max_depth* = 50	0.99377	tcp.srcport, mqtt.dupflag mqtt.len, mqtt.qos mqtt.topic
Random forest	*max_depth* = 50 *n_estimators* = 100	0.99377	tcp.srcport mqtt.dupflag mqtt.len, mqtt.qos mqtt.topic
Ada boost	*n_estimators* = 150	0.97740	tcp.srcport mqtt.dupflag mqtt.len, mqtt.msgtype mqtt.retain, mqtt.topic_len
Gradient boosting	*max_depth* = 25 *n_estimators* = 75 *Learning_rate* = 0.1 *Loss* = deviance	0.99373	tcp.srcport mqtt.dupflag mqtt.len, mqtt.qos mqtt.topic
Log regression	*solver* = liblinear *C* = 1	0.91531	tcp.srcport, frame.cap_len mqtt.conack.flags.reserved mqtt.dupflag, mqtt.len mqtt.msgtype, mqtt.retain mqtt.topic_len
SVC	*kernel* = rbf *C* = 5	0.99023	tcp.srcport, mqtt.dupflag mqtt.len, mqtt.qos mqtt.topic

Table 5 Models performance and selected features for Man-in-the-middle

Model	Best hyper-parameters	Accuracy	Selected features
Decision tree	*max_depth* = 10	0.95902	tcp.srcport, frame.cap_len mqtt.hdrflags, mqtt.msg mqtt.topic
Random forest	*max_depth* = 10 *n_estimators* = 50	0.96094	tcp.srcport frame.cap_len mqtt.hdrflags mqtt.msg, mqtt.topic
Ada Boost	*n_estimators* = 200	0.95838	tcp.srcport frame.cap_len, mqtt.hdrflags mqtt.msg
Gradient boosting	*max_depth* = 10 *n_estimators* = 25 *learning_rate* = 0.01 *loss* = deviance	0.96030	tcp.srcport frame.cap_len mqtt.hdrflags mqtt.msg
Log regression	*solver* = sag *C* = 0.5	0.87580	tcp.srcport, frame.cap_len mqtt.len
SVC	*kernel* = rbf *C* = 2	0.96094	tcp.srcport, frame.cap_len mqtt.hdrflags, mqtt.msg

Table 6 Models performance and selected features for intrusion

Model	Best hyper-parameters	Accuracy	Selected features
Decision tree	*max_depth* = 15	0.95384	tcp.srcport, frame.cap_len mqtt.conack.flags.reserved mqtt.dupflag, mqtt.len mqtt.msgtype mqtt.retain mqtt.topic_len
Random forest	*max_depth* = 25 *n_estimators* = 50	0.95294	tcp.srcport, frame.cap_len mqtt.conack.flags.reserved mqtt.dupflag, mqtt.len mqtt.msgtype, mqtt.retain mqtt.topic_len
Ada boost	*n_estimators* = 200	0.92941	tcp.srcport, frame.cap_len mqtt.conack.flags.reserved mqtt.dupflag, mqtt.len mqtt.msgtype, mqtt.retain mqtt.topic_len
Gradient boosting	*max_depth* = 10 *n_estimators* = 50 *Learning_rate* = 0.1 *Loss* = exponential	0.95384	tcp.srcport, frame.cap_len mqtt.conack.flags.reserved mqtt.dupflag, mqtt.len mqtt.msgtype, mqtt.retain mqtt.topic_len
Log regression	*solver* = liblinear *C* = 5	0.74841	tcp.srcport, frame.cap_len mqtt.conack.flags.reserved mqtt.dupflag, mqtt.len mqtt.msgtype, mqtt.retain mqtt.topic_len
SVC	*kernel* = rbf *C* = 1	0.92941	tcp.srcport, frame.cap_len, mqtt.clientid mqtt.conack.flags, mqtt.conack.flags.sp mqtt.conflag.cleansess, mqtt.conflag.qos mqtt.conflag.retain mqtt.conflag.willflag mqtt.dupflag, mqtt.kalive, mqtt.msg mqtt.proto_len, mqtt.qos, mqtt.topic mqtt.ver

Regarding the possibility to enrich or improve an IDS with machine learning techniques, we can look at the classifiers obtained as a by-product of the feature selection algorithm run over the constructed dataset. For a DoS attack, it is enough to consider 5 features to be able to train a DT classifier with an accuracy of 0.99377. In case of a MITM attack, we just need also 5 features to train a RF classifier with an accuracy of 0.96094. Finally, we can train a GB classifier considering only 8 features with an accuracy of 0.95384. This is summarized in Table 9.

Concerning the comparison of the results of this work with those found in the literature, it is worth pointing out the difficulty to establish a statistical meaningful contrast due to the absence of other analyses on this new dataset as well as to the nonappearance of other works studying and characterizing exactly the same type of attacks. However, we find that one of the most studied type of attack is DoS [15,61,63], which is also considered in this work. In Table 10 we include the existing literature about the construction of IoT security datasets and/or with a machine learning analyses on them. For each reference, we specify the datasets on which the analyses are done, the attacks considered, the number of selected features (complexity of characterization) to perform the analyses and, if possible, the accuracy obtained by the predictors when trying to detect traffic under a DoS attack. If the authors construct predictors over different datasets and with different machine learning models, we include the best accuracy. We also include the corresponding data obtained in this work. As the table shows, our dataset together with the feature selection algorithm provides not only a good characterization of

Table 7 Selected features for each type of attack

	fr.cap_len	**mqtt.connack.flags.reserved**	**mqtt.dupflag**
Man-in-the-middle	✓	×	×
DoS	×	×	✓
Intrusion	✓	✓	✓
	mqtt.len	**mqtt.msgtype**	**mqtt.qos**
Man-in-the-middle	×	×	×
DoS	✓	×	✓
Intrusion	✓	✓	×
	mqtt.retain	**mqtt.topic**	**tcp.srcport**
Man-in-the-middle	×	✓	✓
DoS	×	✓	✓
Intrusion	✓	×	✓
	mqtt.hdrflags	**mqtt.msg**	**mqtt.topic_len**
Man-in-the-middle	✓	✓	×
DoS	×	×	×
Intrusion	×	×	✓

Table 8 Description of selected features

	Description
fr.cap-len	Full frame size
mqtt.connack.flags.reserved	Reserved flag sent by the broker with a new connection
mqtt.dupflag	Flag towards the broker to indicate the attempt to send a MQTT package when publishing a message
mqtt.len	MQTT package size
mqtt.msgtype	Indicates the type of mqtt package, such as subscription publication or disconnection
mqtt.qos	Level of quality of service (0, 1, 2,reserved)
mqtt.retain	Boolean value to indicate if borker stores message as last valid message for a specific topic
mqtt.topic	Indicates the topic where the message is published
tcp.srcport	Random port that changes for each connection
mqtt.hdrflags	Header Flags with package control information
mqtt.msg	The payload of the message
mqtt.topic_len	The size of the topic

Table 9 Models performance and selected features for Intrusion

	DoS	Man-in-the-middle	Intrusion
# features	5	5	8
ML model	DT	RF	GB
Acc	0.99377	0.96094	0.95384

malicious traffic but also a much simpler one. Furthermore, in case of DoS attacks, the simplicity of the characterization does not compromise the classification accuracy obtained by machine learning methods.

It is important to note that it was not the goal of this work to construct classifiers for each type of attack, so this point deserves a deeper and sharper analysis by considering different machine learning techniques.

Conclusions

IoT devices are currently growing exponentially. New security challenges must be addressed due to the special nature of these types of devices, mainly their low cost, which use to lead to security problems. Data Mining and machine learning techniques can be a good way of understanding the behavior of IoT environment vector attacks. However, these techniques require datasets describing certain events and there is a lack of this type of information available to the cybersecurity research community.

Table 10 Summary of available IoT datasets that contain anomalous network traffic

Ref.	Dataset	IoT attacks/classes	# Features	Acc. DoS
[60]	**MQTT-IoT-IDS2020**	Scanning and brute-force/binary and multiclass	64	
[61]	**MQTT-IoT-IDS2020**, **IoT-23**, **MQTT-set**, **Bot IoT**	Scanning and brute-force; botnets; Brute-force, malformed data and DoS; Attacks by generic bots that can affect IoT, such as Dos and DDoS/binary and multiclass	73	0.98500(1)
[43]	**MQTT-IoT-IDS2020**	Scanning and brute-force/multiclass	78	
[63]	**MQTT-set**	Brute-force, malformed data and DoS/multiclass	33	0.84126(2)
[31]	**MQTT-IoT-IDS2020**	Scanning and brute-force/multiclass	29 (Packet) 18 (Unidirectional) 18 (Bidirectional)	
[15]	**Artemis dataset**	DoS attacks on the MQTT network/binary	31	0.99980(3)
–	**Proposed MQTT**	MQTT specifics vulnerabilities attacks/binary	5 (DoS) 5 (MITM) 8 (I)	0.99377

(1) MQTTset dataset binary classification for normal and anomalous classes. (2) Datum obtained from the confusion matrix presented in a multiclass classification with a Multilayer Perceptron. (3) Accuracy obtained by a One-Class Support Vector Machine when JSON objects are included

This work has made two main contributions. First, the development of a procedure for collecting datasets in IoT network environments that works successfully. Second, the design of a hybrid feature selection algorithm that is able to characterize the previous dataset, that contains a large number of network traffic features. Thus, the most important features of Man-in-the-middle, DoS and Intrusion attacks over MQTT IoT protocol have been identified.

The wrapper routine of the hybrid algorithm uses six machine learning classification models. This algorithm returns the most important features together with the aforementioned models trained on the filtered dataset. The high accuracy values obtained by these estimators validate our approach.

Taking these contributions into account, we propose to develop machine learning classifiers trained on this dataset with the selected features to feed IDS's in a way that can detect attacks on the MQTT protocol in an IoT system.

Acknowledgements This work is partially supported by Instituto Nacional de Ciberseguridad (INCIBE), Junta de Castilla y Leon–Consejería de Educación (LE078G18, UXXI2018/000149, U-220), powered by NVIDIA GPU Grant Program and developed in Research Institute of Applied Sciences in Cybersecurity (RIASC).

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. 1998 DARPA Intrusion Detection Evaluation Dataset. MIT Lincoln Laboratory. https://bit.ly/2xseCm8. Accessed 22 Apr 2022
2. Alam F, Mehmood R, Katib I, Albeshri A (2016) Analysis of eight data mining algorithms for smarter internet of things (IoT). Proc Comp Sci 58:437–442
3. Ali A, Shah GA, Farooq MO, Ghani U (2017) Technologies and challenges in developing machine-to-machine applications: a survey. J Netw Comput Appl 83:124–139
4. Aminanto ME, Choi R, Tanuwidjaja HC, Yoo PD, Kim K (2017) Deep abstraction and weighted feature selection for Wi-Fi impersonation detection. IEEE Trans Inf Forensics Secur 13(3):621–636
5. Andy S, Rahardjo B, Hanindhito B (2017) Attack scenarios and security analysis of MQTT communication protocol in IoT system. In: 2017 4th international conference on electrical engineering, computer science and informatics (EECSI), pp 1–6
6. Baig ZA, Sanguanpong S, Firdous SN, Vo VN, Nguyen TG, So-In C (2020) Averaged dependence estimators for DoS attack detection in IoT networks. Futur Gener Comput Syst 102:198–209

7. Bharathi V, Kumar C (2022) Enhanced security for an IoT devices in cyber-physical system against cyber attacks, pp 1–5. https://doi.org/10.1109/ICONAT53423.2022.9725884
8. Breiman L (1998) Arcing classifiers. Ann Stat 26(3):801–849
9. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
10. Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and regression trees. Chapman and Hall/CRC, Boca Raton
11. Brown G, Pocock A, Zhao MJ, Lujan M (2012) Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. J Mach Learn Res 13:27–66
12. Cecchinel C, Jimenez M, Mosser S, Riveill M (2014) An architecture to support the collection of big data in the internet of things. In: 2014 IEEE world congress on services, pp 442–449
13. Chaniotis IK, Kyriakos-Ioannis, Kyriakou D, Tselikas ND, Chaniotis IK, Kyriakou KID, Tselikas ND (2015) Is Node.js a viable option for building modern web applications? A performance evaluation study. In: Computing, vol 97, pp 1023–1044
14. Chen J, Yang T, He B, He L (2021) An analysis and research on wireless network security dataset. In: Proceedings—2021 international conference on big data analysis and computer science, BDACS 2021. Institute of Electrical and Electronics Engineers Inc., pp 80–83. https://doi.org/10.1109/BDACS53596.2021.00025
15. Ciklabakkal E, Donmez A, Erdemir M, Suren E, Yilmaz MK, Angin P (2019) Artemis: an intrusion detection system for mqtt attacks in internet of things. In: 38th symposium on reliable distributed systems (SRDS). IEEE, pp 369–371
16. Cisco: internet of things at a glance (2016). https://www.cisco.com/c/en/us/products/collateral/se/internet-of-things/at-a-glance-c45-731471.pdf?dtid=osscdc000283. Accessed 22 Apr 2022
17. Collina M (2018) Mosca@github.com. https://github.com/mcollina/mosca. Accessed 22 Apr 2022
18. Conti M, Dehghantanha A, Franke K, Watson S (2018) Internet of things security and forensics: challenges and opportunities. Futur Gener Comput Syst 78:544–546
19. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
20. Defazio A, Bach F, Lacoste-Julien S (2014) Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In: Proceedings of the 27th international conference on neural information processing systems—vol 1, NIPS'14. MIT Press, pp 1646–1654
21. Ding C, P H (2005) Minimum redundancy feature selection from microarray gene expression data. J Bioinform Comput Biol 3:185–205
22. Ding C, Peng H (2005) Minimum redundancy feature selection from microarray gene expression data. J Bioinform Comput Biol 3:185–205
23. Do VT, Engelstad P, Feng B, van Do T (2016) Strengthening mobile network security using machine learning. In: Younas M, Awan I, Kryvinska N, Strauss CT (eds) Mobile web and intelligent information systems. Springer International Publishing, Cham, pp 173–183
24. Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 55(1):119–139
25. Friedman JH (1999) Stochastic gradient boosting. Comput Stat Data Anal 38(4):367–378
26. Garg S, Kaur K, Batra S, Kaddoum G, Kumar N, Boukerche A (2020) A multi-stage anomaly detection scheme for augmenting the security in IoT-enabled applications. Futur Gener Comput Syst 104:105–118
27. Garg S, Singh R, Obaidat MS, Bhalla VK, Sharma B (2020) Statistical vertical reduction-based data abridging technique for big network traffic dataset. Int J Commun Syst 33(4):1–13
28. Gerganov R (2018) nfqsed@github.com. https://github.com/rgerganov/nfqsed. Accessed 22 Apr 2022
29. Getting started with mqtt. https://www.hivemq.com/blog/how-to-get-started-with-mqtt. Accessed 22 Apr 2022
30. Handosa M, Gracanin D, Elmongui HG (2017) Performance evaluation of mqtt-based internet of things systems. In: 2017 Winter simulation conference (WSC), pp 4544–4545
31. Hindy H, Bayne E, Bures M, Atkinson R, Tachtatzis R, Bellekens X (2020) Machine learning based iot intrusion detection system: an mqtt case study (mqtt-iot-ids2020 dataset). In: Lecture notes in networks and systems, international network conference. Springer International Publishing, pp 73–84
32. Hua Yang H, Moody J (1999) Feature selection based on joint mutual information. In: Advances in intelligent data analysis (AIDA), computational intelligence methods and applications (CIMA). IEEE
33. Kalimuthan C, Arokia Renjit J (2020) Review on intrusion detection using feature selection with machine learning techniques. Mater Today Proc 33:3794–3802
34. KDD Cup 99 Dataset (1999). http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. Accessed 22 Apr 2022
35. Kolias C, Kambourakis G, Stavrou A, Gritzalis S (2016) Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset. IEEE Commun Surv Tutor 18:1
36. Kolias C, Kambourakis G, Stavrou A, Voas J (2017) DDoS in the IoT: Mirai and other botnets. Computer 50(7):80–84
37. Kolter JZ, Johnson MJ (2011) REDD : a public data set for energy disaggregation research. In: Proceedings of the SustKDD workshop on data mining applications in sustainability, pp 1–6
38. Kvalseth TO (1987) Entropy and correlation: some comments. IEEE Trans Syst Man Cybern 17:517–519
39. Larriva-Novo X, Villagrá VA, Vega-Barbas M, Rivera D, Sanz Rodrigo M (2021) An IoT-focused intrusion detection system approach based on preprocessing characterization for cybersecurity datasets. Sensors (Switzerland) 21(2):1–15. https://doi.org/10.3390/s21020656
40. Ledun J (2016) jledun@github.com. https://github.com/jledun
41. Lohiya R, Thakkar A (2020) Application domains, evaluation datasets, and research challenges of IoT: a systematic review. IEEE Internet Things J 4662:1–25
42. Maciá Fernández G, Camacho J, Magán-Carrión R, García-Teodoro P, Theron R (2018) UGR 16: a new dataset for the evaluation of cyclostationarity-based network IDSs. Comput Secur 73:411–424
43. Mosaiyebzadeh F, Araujo Rodriguez LG, Macedo Batista D, Hirata R (2021) A network intrusion detection system using deep learning against mqtt attacks in iot. In: Younas M, Awan I, Kryvinska N, Strauss C, Thanh DV (eds) 2021 IEEE Latin-American conference on communications (LATINCOM). IEEE, pp 1–6
44. Mosquitto E index@mosquitto.org. https://mosquitto.org/. Accessed 22 Apr 2022
45. Nimbalkar P, Kshirsagar D (2021) Feature selection for intrusion detection system in internet-of-things (IoT). ICT Express 7(2):177–181. https://doi.org/10.1016/j.icte.2021.04.012
46. Nobakht M, Sivaraman V, Boreli R (2016) A host-based intrusion detection and mitigation framework for smart home iot using openflow. In: 2016 11th International conference on availability, reliability and security (ARES), pp 147–156
47. OASIS (Organization for the Advancement of Structured Information Standards): MQTT Version 3.1.1. Tech. Rep. October (2014). http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html
48. Palsson K (2018) mqtt-malaria@github.com. https://github.com/remakeelectric/mqtt-malaria. Accessed 22 Apr 2022
49. Pa YMP, Suzuki S, Yoshioka K, Matsumoto T, Kasama T, Rossow C (2016) IoTPOT: a novel honeypot for revealing current IoT threats. J Inf Process 24(3):522–533

50. Ponnusamy V, Yichiet A, Jhanjhi NZ, Humayun M, Almufareh MF (2021) IoT wireless intrusion detection and network traffic analysis. Comput Syst Sci Eng 40(3):865–879. https://doi.org/10.32604/CSSE.2022.018801
51. Rahman MA, Asyhari AT, Wen OW, Ajra H, Ahmed Y, Anwar F (2021) Effective combining of feature selection techniques for machine learning-enabled IoT intrusion detection. Multimedia Tools Appl 80(20):31381–31399. https://doi.org/10.1007/s11042-021-10567-y
52. Razzaq MA, Habid S, Ali M, Ullah S (2017) Security issues in the internet of things (IoT): a comprehensive study. Int J Adv Comput Sci Appl 8(6):383
53. Schmidt M, Le Roux N, Bach F (2017) Minimizing finite sums with the stochastic average gradient. Math Program 162:83–112
54. Sebbana M, Nockb R (2002) A hybrid filter/wrapper approach of feature selection using information theory. Pattern Recogn 35:835–846
55. Siddique K, Akhtar Z, Aslam Khan F, Kim Y (2019) KDD Cup 99 data sets: a perspective on the role of data sets in network intrusion detection research. Computer 52(2):41–51. https://doi.org/10.1109/MC.2018.2888764
56. Stanislav M, Beardsley T (2015) Hacking iot: a case study on baby monitor exposures and vulnerabilities. Rapid 7:17
57. STARR M Fridge caught sending spam emails in botnet attack. https://cnet.co/33exaly. Accessed 22 Apr 2022
58. Tsai CF, Hsu YF, Lin CY, Lin WY (2009) Intrusion detection by machine learning: a review. Expert Syst Appl 36(10):11994–12000
59. Tsamardinos I, Aliferis CF, Statnikov AR (2003) Algorithms for large scale Markov blanket discovery. In: FLAIRS conference
60. Ullah I, Mahmoud QH (2021) Design and development of a deep learning-based model for anomaly detection in iot networks. IEEE Access 9:103906–103926
61. Ullah I, Mahmoud QH (2021) A framework for anomaly detection in iot networks using conditional generative adversarial networks. IEEE Access 9:165907–165931
62. Usha M, Kavitha P (2017) Anomaly based intrusion detection for 802.11 networks with optimal features using SVM classifier. Wireless Netw 23(8):2431–2446
63. Vaccari I, Chiola G et al (2020) Mqttset, a new dataset for machine learning techniques on mqtt. Sensors (Basel) 20:6578
64. Wurm J, Hoang K, Arias O, Sadeghi AR, Jin Y (2016) Security analysis on consumer and industrial IoT devices. In: 2016 21st Asia and South Pacific design automation conference (ASP-DAC), pp 519–524
65. Xuan Vinh N, Epps J, Bailey J (2010) Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. J Mach Learn Res 11:2837–2854
66. Yu H, Huang F, Lin C (2011) Dual coordinate descent methods for logistic regression and maximum entropy models. Mach Learn 85:41–75
67. Zarpelão BB, Miani RS, Kawakani CT, de Alvarenga SC (2017) A survey of intrusion detection in internet of things. J Netw Comput Appl 84(February):25–37
68. Zhou D, Yan Z, FU Y, Yai Z (2018) A survey on network data collection. J Netw Comput Appl 116:9–23