Features-Aware DDoS Detection in Heterogeneous Smart Environments based on Fog and Cloud Computing

Wanderson L, Costa[1], Ariel L. C Portela[1] and Rafael L. Gomes[1]

[1]State University of Ceará , Brazil

***Abstract*:** Nowadays, urban environments are deploying smart environments (SEs) to evolve infrastructures, resources, and services. SEs are composed of a huge amount of heterogeneous devices, i.e., the SEs have both personal devices (smartphones, notebooks, tablets, etc) and Internet of Things (IoT) devices (sensors, actuators, and others). One of the existing problems of the SEs is the detection of Distributed Denial of Service (DDoS) attacks, due to the vulnerabilities of IoT devices. In this way, it is necessary to deploy solutions that can detect DDoS in SEs, dealing with issues like scalability, adaptability, and heterogeneity (distinct protocols, hardware capacity, and running applications). Within this context, this article presents an Intelligent System for DDoS detection in SEs, applying Machine Learning (ML), Fog, and Cloud computing approach. Additionally, the article presents a study about the most important traffic features for detecting DDoS in SEs, as well as a traffic segmentation approach to improve the accuracy of the system. The experiments performed, using real network traffic, suggest that the proposed system reaches 99% of accuracy, while reduces the volume of data exchanged and the detection time.

***Keywords*:** Machine Learning, DDoS Detection, Features Selection, Security System**.**

1. Introduction

The human society are claiming for more intelligent urban environments that deploy services for the end users. This kind of environment has been called a Smart Environments (SEs), that can be implemented in several contexts: Smart Campus, Smart Homes, Smart Cities, Industry 4.0, Smart Hospitals, etc [7]. These contexts have singular services to evolve the quality of the life of the end users.

SEs are composed of Internet of Things (IoT) devices (like sensors and actuators) and personal devices (such as notebooks, smartphones, tablets, etc) [15]. Hence, SEs have two crucial characteristics: Huge amount of devices and Heterogeneity. As a consequence, SEs tend to produce more network flows than traditional networks, due to the enormous scale of devices in the network, as well as the various types of applications running on the top of these devices. All these issues rise new challenges related to the management and planning of the SEs and their services [2].

One of these challenges of SEs is the detection of Distributed Denial of Service (DDoS) attacks, that aim to make access to one or more targets unavailable by exhausting their resources using multiple illegitimate requests. The DDoS attacks come from numerous security vulnerabilities in the devices, specially IoT devices [3,6], that directly affect the Quality of Service (QoS) and Quality of Experience (QoE). As a result, in the last few years, several cyberattacks performed in the Internet occurred through the infection of IoT devices [4].

The IoT devices hardware limitations prevent the deployment of security solutions that run on them. An alternative approach is the usage of Machine Learning (ML) techniques, which understand the available data behavior and progressively improve the understanding of them [7]. However, it is necessary to use the most relevant characteristics of network traffic to later train the DDoS attack detection mechanism using ML, since the consideration of unsuitable characteristics harms the accuracy of ML techniques.

Moreover, the monitoring of this network flow generates a high volume of data, making the application of Fog and Cloud computing essential to this scenario [19,1]. Traditionally, Cloud Computing is applied in scenarios of high processing demand, due to the virtually limitless storage and processing capability [8]. Nevertheless, the transmission of raw data related to the network flows to the cloud generates overhead in the network infrastructure as a whole. The addition of a Fog Computing environment between the SE and the Cloud enables the processing of the raw data and the reduction of data volume exchange. This structure increases the performance and privacy of the data in the IoT, as well as reduces the latency [19].

Within this context, this article presents an Intelligent System for DDoS detection in SEs, which integrates Fog and Cloud, splitting the tasks in these two computing environments to reduce the response time and to improve the accuracy. The proposed system is based on the following principles: (I) Network monitoring, collection of data about the network flows in the SE; (II) Features Selection, identification of the main characteristics for the detection of DDoS in SEs; (III) Traffic Segmentation, separation of network flows from IoT devices and Personal devices; and, (IV) ML for Detection, training of ML models using the data about the network flows to detect DDoS attacks.

The experiments performed, using a dataset of real IoT network traffic with DDoS attacks, suggest that the proposed system reaches 99% of accuracy when the most suitable features are selected, while reducing the volume of data exchanged and the detection time.

This article has the following contributions: (A) Design of an system to integrate cloud and fog computing; (B) Study about the impact of the features selection under the DDoS detection accuracy, as well as the amount of data exchange (traffic volume) between cloud and fog; (C) Traffic segmentation approach to separate the network flows of IoT devices from the Personal devices in the SE, which could be applied in another task for the management of SEs (such as authentication, firewall, etc). And, (D) Experiments using a dataset of real network traffic with DDoS attacks.

The remainder of this article proceeds as follows. Section 2 presents related works. Section 3 describes the proposed system. Section 4 details the performance evaluation and results. Finally, Section 5 concludes the article.

2. Related Works

Hamamoto et al. [12] proposed a scheme based on the combination of genetic algorithm and fuzzy logic. The learning structures work together to network traces created by the genetic algorithm. The fuzzy logic defines when the network is normal or under a cyberattack from a previously generated signature. Despite based on real traffic and AI techniques, the proposed scheme focuses on traditional networks, i.e., the authors do not consider SE characteristics.

Vinayakumar et al. [22] propose a botnet detection system based on a two-tier ML structure to semantically distinguish botnets from legitimate behavior in the application layer of DNS domain name system services. In the first level, scores are used to define the similarity, when reaching a difference established by the authors, the domain name is passed to the second level that uses a deep learning architecture to detect and classify the occurrences of DDoS. This work focuses on the detection of DDoS exclusively on DNS servers, preventing its application in other types of IoT network services.

Sharafaldin et al. [20] present a study on the traffic characteristics of the most important networks for detecting different types of DDoS attacks on traditional networks, that is, TCP / IP networks. In the carried out experiments, two networks with traditional computers were designed and implanted, that is, the behavior extracted from the dataset samples becomes different in comparison with networks designed with IoT devices. The behavior of IoT networks communicates with a small finite set of endpoints and is prone to have repetitive network traffic patterns (small packages at fixed time intervals for registration purposes, for example).

Yamauchi et al. [24] describe a model for detecting anomalous operations of IoT devices in smart homes (SHs) based on user behavior. The model learns the sequence of activities performed by hour of the day and then compares the current sequence with the sequences learned for the condition corresponding to the current condition. If it has any predefined changes, the method classifies the operation as an IoT device anomaly. Thus, this model proposed by the authors is limited to understanding SHs.

Doshi et al. [7] performed the detection of ongoing DDoS attacks through the IoT flow behavior in smart homes. The approach deploys middleboxes, acting as proxies in the network to observe, store, process and control network traffic going to the Internet. This strategy monitors flow characteristics, such as inter-packet arrival time, endpoints and other. The collected information serves as input to a ML technique to create a model to identify possible DDoS bot in IoT, i.e., a binary classification (DDoS bor or Safe node). The authors evaluated several ML techniques: KNN, Lagrangian Support Vector Machine (LSVM), Decision Tree (DT), Random Forest (RD) and Neural Networks. However, the approach is specific for DDoS attacks.

HaddadPajouh et al. [11] explored the application of Recurrent Neural Network (RNN) deep learning in detecting IoT malwares. First, the authors collected IoT malware and benign samples to build a dataset. Later, the authors use RNN to analyze ARM-based IoT applications' execution operation codes (OpCodes), creating a feature vector file based on the OpCodes for each sample. In the final stage, they utilized vectored data for deep neural network training and tuning for optimum parameters. The evaluation of the trained model was based on distinct IoT malware samples, resulting in an accuracy of 98.18% with 2-layer neurons. Nevertheless, the trained model depends of the analysis of OpCodes, limiting its capacity to detect Mirai Botnet.

Zhou et al. [25] presented an Intrusion Detection System (IDS), which applies an heuristic algorithm, called Correlation-based Selection Bat Algorithm (CSBA). CSBA supports the measurement of correlation between network resources, selecting the most suitable subset of data to perform the training of ML techniques (Random Forest and Forest PA). The CSBA algorithm reduces the training time from 113 minutes to 44 minutes. Regarding the accuracy, the proposed IDS reached small false alarm rates, around 0.17%. Nevertheless, the proposed IDS can not deal with the network, avoiding its application in a smart environment. Another limitation is the lack of concern about the processing capacity and the volume of traffic generated to perform the detection. All these issues hamper the deployment of this solution in SEs.

Diro et al. [6] and Brun et al. [4] proposed an attack detection system and designed an architecture, respectively, based on deep learning for IoT, comparing this technique with other existing machine learning approaches. The authors consider a cloud structure, where the information collected in distinct IoTs are received and processed in a master node of the cloud. The system evaluation is employed as input data from NSL-KDD, considering the centralization of all collected data in the cloud. The authors did not include any strategy to avoid the transmission overhead from data exchange.

In [4], the deep learning model uses information about the message exchange between IoT devices and IoT gateway, and the IoT gateway and Cloud. All packets are sent to the deep learning training module at the Cloud. The architecture evaluation does not consider the volume of data exchanged to allow the model training, i.e., it does not prevent the transmission overhead.

Meidan et al. [16] present an unsupervised patterns approach using Deep Autoencoders to detect botnets in IoT networks. The authors suggest that only twenty-three characteristics for training the learning method is enough to reach suitable accuracy. The experiments were performed in a testbed composed of IoT devices. Nevertheless, this work is specific for botnet detection and it does not consider the processing capacity of the devices and other existing limitations of SEs.

To the best of our knowledge, there is no proposal in the literature focused on the design of an intelligent system to detect DDoS in heterogeneous smart environments integrating Fog, Cloud and Machine Learning techniques, which is the focus of this article. These features create a suitable detection system that evolves the security and management capacity of smart environments, while mitigating the transmission of high volume of data and response time.

3. Proposal

Smart environments are composed of heterogeneous devices, such as sensors, actuators, smartphones, tablets, smart TVs, smartbands and others [2]. Each of these devices follows specific functionalities and, consequently, singular network behavior for a certain class/type of device. For example, temperature sensors perform periodical transmissions for a server, updating a set of data. Similarly, surveillance cameras constantly transmit captured images. Personal devices follow a less predictable behavior: in a given moment the user is just sending text messages (small volume of traffic) and in another moment this same user is watching an on-demand video in high definition (very high volume of traffic).

These characteristics of smart environments increase the management complexity and, consequently, the development of security solutions, due to the limitations of the devices (processing power, energy consumption, etc). One of the most important security solution is the detection of DDoS attacks. Based on this, this article proposes a Intelligent DDoS Detection system using ML techniques and supported by Fog and Cloud Computing.

The proposed system performs the following stages: (I) Network Monitor; (II) Features Extraction; (III) Traffic Segmentation; (IV) Selection of Features; (V) Knowledge Dataset Formation; (VI) Training of Model; and, (VII) Detection of DDoS. An overview of the proposed system structure is presented in Figure 1.

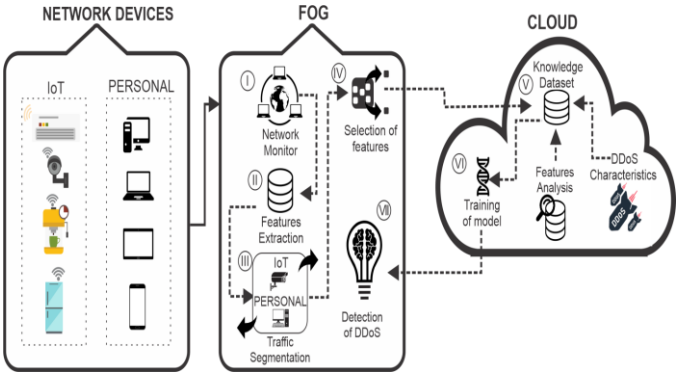

Figure 1. Overview of the Proposed System

Additionally, the development of the proposed system encompasses two pre-processing steps to build the basic knowledge: Reasoning about DDoS, a research to perceive the important characteristics of the attack execution; and, Analysis of Traffic Features, a study to identify the most important features to improve the accuracy of DDoS detection.

The network monitoring is performed, creating an information database. From this raw data in the database, the possible features about the network traffic are extracted. In possession of all this information, traffic segmentation will be executed and features will be selected. These selected features are used to feed a Knowledge dataset, which is applied as input to train the ML model. After training, the generated classifier acts to detect DDoS attacks.

During the features selection and model training phases, various techniques can be used. Based on this, during the development of the proposed system, several techniques were applied and tested. Regarding the Fog and Cloud Computing support, the proposed system is designed to split the processing of the stages. The Fog performs the raw data processing to feed the Knowledge Dataset and, consequently, the Training of Mode located in the Cloud.

All the stages in the data processing flow of the proposed system are illustrated in Figure 2, where the tasks performed and the techniques considered (in features selection and model training) are highlighted. These stages execute sequentially, exchanging data between Fog and Cloud, according to the structure presented in Figure 1.

Next, we describe the stages in the data processing flow, detailing their particularities, as well as the role of each stage for the functioning of the proposed intelligent system as a whole.

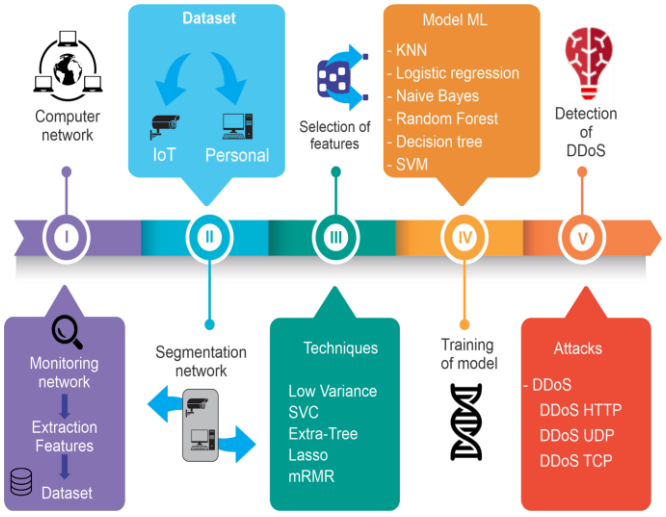

Figure 2. Data Processing Flow

3.1. Reasoning about DDoS

DDoS became a popular cyberattack in the last few years, mainly due to the growth of heterogeneous devices in the networks. The DDoS occurs when a malicious agent (called master) enslaves several devices, forcing these devices to congest an target inside or outside the network [23]. Usually, DDoS can target both applications and the network infrastructure:

- Applications: DDoS affects the services running in the top of the SE, where packets are dropped when the maximum processing rate is achieved. In this case, the goal of the DDoS is to make the service unavailable.
- Network Infrastructure: DDoS focus on the exploration of the vulnerabilities in the network protocols (mainly Transport, Network and Data Link layers). Thus, the DDoS aims to make the service inaccessible by the legitimate users.

From the reasoning performed, it is possible to define the possible approaches to detect DDoS in SEs. Initially, a security solution needs to get useful information from the network packets (such as device addresses, used protocols, signaling flags, and others). Additionally, it is necessary to use the information about the correlated packets transmitted through the network infrastructure, i.e., network flows from the devices to the gateway (for example, inter packet interval, volume of data, average packet size, etc). These information are important to model the behavior of the network flows and, consequently, the profile of the devices.

It is important to note that a suitable security solution should not be based on previous information about the devices, since SEs are characterized by a huge amount of heterogeneous and mobile devices. This fact turns the collection of data about the network and the processing of it crucial tasks to get useful information. In this way, the proposed system was based on the suitable data processing (considering the best features about the data to be used and the processing time) to perform the detection of DDoS attacks in the SE.

3.2. Extraction of Features

Based on reasoning about DDoS attacks, the proposed system extracts 80 (eighty) characteristics of network flows and packets, according to the standard of the CICFlowMeter tool [20] (which extracts features from raw data in PCAP format). The following are examples of the features extracted: application layer, network and link protocols; information in

the packet header (such as flags, version, etc), both total and payload size of the packets, inbound and outbound volume of a flow, inter-packet time, and others.

3.3. Segmentation of Network Flows

Real-life network architectures in an SE are a wide variety of IoT and personal devices, where they communicate with each other and with services outside of the SE through the Internet. As a consequence, IoT and personal devices have distinct behaviors: IoT devices are fixed, while personal devices are mobile; IoT devices have specific applications, while personal devices have a pool of applications with different characteristics (text messaging, videos on demand, games, etc); and so on. Together with this fact, IoT devices have numerous security vulnerabilities [3,6]. For instance, several DDoS attacks performed in the Internet occurred through the infection of IoT devices [4].

In this way, the proposed intelligent system performs a traffic segmentation approach, i.e., it identifies the network flows of IoT devices from Personal devices, allowing the training of the ML technique using the data according to the type of device. As a result, it is possible to better fit the behavior of each type and, consequently, improve the detection of DDoS attacks.

The traffic segmentation uses ML to classify IoT and Personal devices based on the features extracted from the monitoring of the network flows. It explores the distinctive characteristics of the devices when they communicate in an SE. The usage of ML allows the traffic segmentation to be adaptable to new devices in the SE, which occurs due to mobility and expansion of services running on it.

For the development of the traffic segmentation, we used a existing dataset[1] (developed by Meidan et al. [17,14]) of different IoT devices (such as covering cameras, lights, plugs, motion sensors, devices, health monitors, among others) to train the ML model. In Section 4.2.1, we presented the results about traffic segmentation, evaluation of the KNN, Logistic regression, Naive Bayes, Random Forest, Decision tree and SVM techniques.

3.4. Selection of Features

All the information extracted represents some aspect of IoT Networks. However, the use of a large number of features will result in certain noises that may interfere with the process of ML model. In addition, noise affects the functioning of ML techniques unevenly, that is, a certain noise may or may not affect another technique. Therefore, it is important to select the most relevant characteristics in order to achieve the best performance of the classifiers used in DDoS detection.

The selection of characteristics for network traffic analysis is a challenge for specialists who aim to build systems that discover patterns of behavior. This process becomes even more complex when it comes to DDoS attacks due to the variety of types, as well as the complexity of its action timing. The selection purpose of characteristics is to enable the construction of ML models that make it possible to understand the data and maximize the detection capacity. Therefore, the selection of characteristics helps to know irrelevant and redundant attributes that can have a negative impact on the model performance, decreasing the accuracy of the model.

In addition, reducing the number of features brings important benefits when looking at computational resources. Less data means reduced training time, less misleading data that improves model performance, faster processing, less memory consumption, easier data extraction, less storage space and, mainly, dimensionality reduction. Thus, the appropriate characteristics selection makes it possible to optimize the time for training and detection of these ML models.

Based on these facts, this articles analyzes the following techniques for selecting characteristics: (1) Maximum relevance Minimum Redundancy (mRMR) [18], uses Fisher's test scores and Pearson's correlation; (2) Low Variance (LV), removes all characteristics whose variation does not reach a certain limit; (3) Extra-Tree (EA) [10], builds a set of non-pruned decision or regression trees according to the classic top-down procedure; (4) SVC [5], a linear model that estimates sparse coefficients based on important characteristics; and, (5) Lasso [9], a linear model that estimates sparse coefficients.

Any of these techniques can be used to select the relevant characteristics of the DDoS data set that improves the models performance. Nevertheless, the different strategies applied by them (filter methods, wrapping methods or embedded methods) lead to different selected characteristics [13].

In this way, we present in Section 4 a study to define the most suitable selection technique that improves the accuracy of the existing ML models to detect DDoS in SEs. Additionally, timing issues are evaluated: Training time and detection time. This timing evaluation allows the identification of the tuple selection technique together with the ML model that is faster, allowing its application in time constrained scenarios.

3.5. Model Training and Detection of DDoS Attacks

After transmitting the processed data from the Fog to the Cloud, the Knowledge Dataset is fed and it is used as the basis for the ML training. ML training encompasses the input of the data in the Knowledge Dataset and the execution of the ML technique. Later, the detector (ML model trained) is transmitted and executed in the Fog to identify possible DDoS attacks.

The proposed system was designed to enable the usage of any ML technique. This Independence allows the proposed system to execute the most suitable ML technique in the training stage. Thus, we evaluated the ML techniques that have distinct singularities: K-Nearest Neighbor (KNN), Naive Bayes (NB), Random Forest (RF), Decision Tree (DT), Logistic Regression (LR) and Support Vector Machines (SVM).

The defined process flow is repeated constant to keep the detector updated according to the behavior of the devices in the SE. This continuous feedback process enables the recurrent information knowledge about the smart environment. As a consequence, the proposed solution is capable of adapting and to understand the usual behavior the network flows and to detect DDoS attacks.

3.6. Fog and Cloud Support

Data stream generated by the network monitoring needs to be processed before the transmission to the Cloud. Currently, the cloud is the usual place for executing services. Nevertheless, with the ever-growing scale of the network flows of smart environments and, consequently, the volume of the data streams generated by the devices, can create an enormous transmission overhead to the Internet.

[1] https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/bot_iot.php

According to [19], in the next few years, the Internet infrastructure will face the challenge of handling a rise in resource demand due to the data stream of the emerging networks, reaching the order of petabytes every day. In this scenario, approaches such as sending all raw data to be processed and stored in the cloud are impractical in terms of communication time, financial cost, performance degradation and energy consumption. An approach to deal with these issues is not sending all data to be processed by the cloud (far from the data source), deploying a Fog Computing approach. The inclusion of a Fog between the smart environment and the Cloud enables the processing, communication and temporary storage near the smart environment. Therefore, the Fog naturally increases the performance, security and privacy in the smart environment, as well as reduces data volume and the latency.

Furthermore, the ML techniques applied in the DDoS detection demand a high level of computational resources (parallel processing and memory capacities). In general, these computational resources are not available in the Fog, requiring several services to support access networks. Hence, the execution of all the functionalities of the proposed intelligent system in the Fog environment is not feasible.

Collected Raw data is sent to the Fog Environment to be processed, performing the steps described in Sections 3.2, 3.3 and 3.4. After these steps, the raw data turns into processed data, which will be sent to the cloud. This processing reduces data volume, since only useful information is considered to define the processed data.

After Data Processing, data is available in the Cloud environment and it is used as input of the ML techniques to train the DDoS detector. As a final step, the detector is deployed in the Fog environment. Thus, the processed data has two roles: (a) feed the ML training in the cloud and (b) be tested by the DDoS detector in the Fog.

It is worthy to mention that the designed structure supports the application of the proposed system in several Smart Environments that share the same edge network and Fog Environment, where the Cloud Environment will instantiate a virtual machine for each Smart Environment monitored.

In summary, the designed structure of the proposed intelligent system enables two important features: (1) Small overhead in the network infrastructure, due to the low volume of data transmitted between the Fog and Cloud environments; and, (2) Suitability of execution, since each step of the modules runs in the suitable environment, i.e., the ML techniques are executed in the cloud, while the data processing runs in the Fog. These two features allow the system to deal with scalability, adaptability and response time requirements of the smart environments [19,1,8,2,26].

The benefits of the Fog and Cloud structure in the proposed system are evaluated in Section 4, where the reduction of volume of data transmitted from the Fog to the Cloud are analyzed, according to the features selection performed (detailed in Section 3.4).

4. Experiments

This section presents the experiments performed to evaluate the proposed intelligent system for network anomaly detection in smart environments. The experiments focus on the evaluation of the designed Fog-Cloud approach, as well as on the analysis of the most suitable ML technique to detect DDoS attacks.

4.1. Experiments Configuration

During the experiments, the following selection techniques were evaluated: Extra-Tree (EA), SVC, Lasso, Low Variance (LV) and mRMR (cases of 5, 10, 20, 30 and 40 features). These techniques selected the features to be used in the ML techniques trained: DT, SVM, KNN, RF and LR (The complete list of the features selected by each technique is available in Appendix A). Therefore, we evaluated all the possible combinations of selection and ML techniques, allowing a complete analysis about the possible performances. Regarding the hardware used in the experiments, the Fog executed in a local machine with Linux, CPU Intel i7-8700k 2666mhz and 16GB of Memory RAM DDR4, while the Cloud was a F48s-V2 Azure virtual machine with 48 3.4GHz vCPUs and 96GB of Memory RAM. Thus, we performed the experiments in suitable Fog and Cloud environments for realistic scenarios.

The experiments were based on two datasets that were merged to represent an SE composed of heterogeneous IoT and Personal devices. The former is the dataset BoT-IoT[2] developed by Meidan et al. [17,14], which contains both normal (benign) traffic and traffic related to the latest DDoS attacks. The latter is the "UNSW-IoT" created by Sivanathan et al. [21], that has normal (benign) traffic of IoT and Personal devices. Both datasets are formatted in real world monitoring data (PCAPs).

The performance of the proposed intelligent system (including the combination of selection and ML techniques), considering the cases of True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) for a DDoS detection, was based on the following evaluation metrics:

- Accuracy (in percentage): Rate of correct classification, regardless the class, according to the Equation 1. It is important to note that the Accuracy was measured for the Traffic Segmentation and the DDoS detection.

$$ACC = \frac{TP + TN}{\mathrm{TP} + FN + FP + TN} \quad (1)$$

- Recall (in percentage): Efficiency of the classifier to detect the correct class, i.e., the rate of TP in relation to total positive cases (TP+FN). Thus, the Recall is defined in Equation 2.

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + FN} \quad (2)$$

- Training Time (in seconds): time required to train the DDoS detector (ML model) with the selected input features.
- Detection Time (in seconds): time spent by the DDoS detector to define whether a case is a DDoS attack or not.
- Volume of Data (in Megabits): the size of the data generated (processed data) to be exchanged between Fog and Cloud.

4.2. Results

4.2.1 Traffic Segmentation

In this section, we evaluate the capacity of the ML techniques used to identify the IoT devices in the network, that are

[2] https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/bot_iot.php

presented in Figure 3. It is possible to note that DT and RF achieve the best results, reaching an accuracy close to one hundred percent.

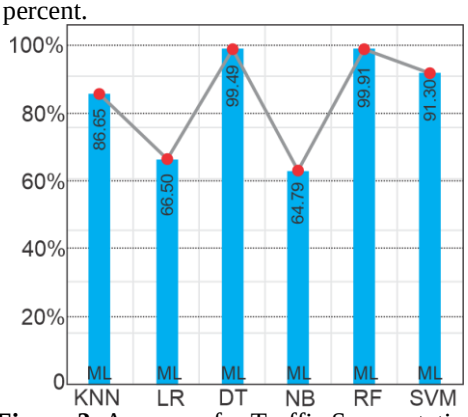

Figure 3. Accuracy for Traffic Segmentation

These better results of DT and RF occur due to their nature to split the problem in multiple stages. In this way, the dual possibility of the classification (IoT or Personal device) eases the division of the problem, improving the organization of the leafs and structure of the designed classification tree.

4.2.2 Detection Accuracy and Recall

The Accuracy was divided in Figures 4 and 5 to facilitate the visualization of the results, where Figure 5 shows the accuracy of the ML techniques using the cases of mRMR and Figure 4 the remaining combinations. Additionally, the Recall results present a similar behavior.

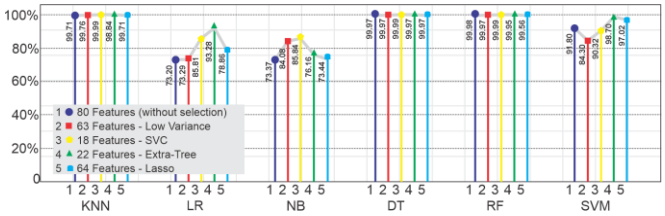

Figure 4. Accuracy for DDoS Detection

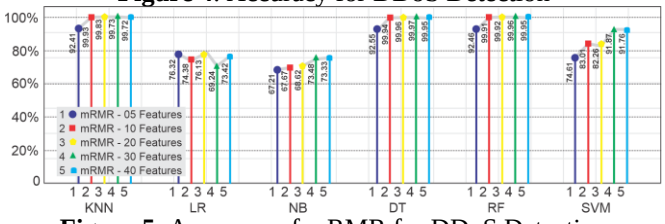

Figure 5. Accuracy of mRMR for DDoS Detection

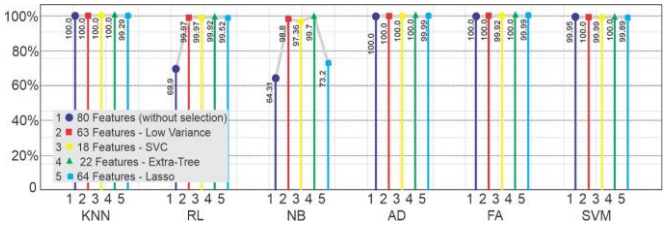

Figure 6. Recall for DDoS Detection

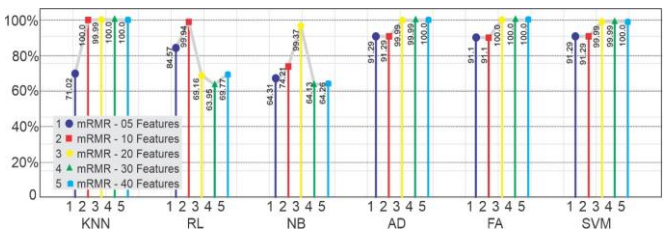

Figure 7. Recall of mRMR for DDoS Detection

From the results shown in the figures, it can be seen that the accuracy and the recall of the ML techniques varies according to the applied selection technique, especially when these ML techniques are based on approaches that focus on dimensionality, such as KNN, RL and SVM classifiers. On the other hand, the ML techniques based on subset division (DT and RF) have almost no impact by the variation in selection techniques. It happens because of the recursive derivation process of the subsets, mitigating the variation on the performance of features and resulting in possible noises for the ML training.

Regarding NB and LR performance, both ML techniques present worse results than the other approaches, regardless of the selection technique. Thus, NB and LR appear as unsuitable solutions for DDoS detection in SEs when compared to the other approaches of the experiment.

4.2.3 Training and Detection Time

Tables 1 and 2 show the time spent to perform the ML model training (creating the DDoS detector) and for the detectors to identify the cases of DDoS attacks, respectively. The results presented in both tables represent the feasibility of the ML techniques to be deployed in distinct contexts of SEs.

Table 1. Training Time (in seconds)

Technique	KNN	LR	NB	DT	RF	SVM
80 Features	13.29	2.34	0.53	1.96	22.99	1625.04
LV	11.74	2.14	0.41	1.88	10.10	273.34
SVC	36.30	1.27	0.21	0.28	5.27	1226.96
Extra-Tree	18.55	2.96	0.19	0.68	13.33	351.01
Lasso	11.91	1.20	0.14	0.74	6.73	304.38
mRMR 05	21.78	2.91	0.13	0.16	4.11	1055.70
mRMR 10	16.01	3.43	0.14	0.38	5.45	708.88
mRMR 20	10.67	3.42	0.30	0.81	10.12	1177.78
mRMR 30	7.91	0.93	0.29	0.49	10.68	385.93
mRMR 40	9.16	1.23	0.28	1.19	15.53	467.02

Table 2. Detection Time (in seconds)

Technique	KNN	LR	NB	DT	RF	SVM
80 Features	9.02	0.02	0.53	0.02	0.53	162.57
LV	7.98	0.03	0.03	0.01	0.34	15.24
SVC	15.62	0.02	0.02	0.01	0.35	144.13
Extra-Tree	12.95	0.01	0.06	0.02	0.44	21.15
Lasso	8.15	0.01	0.14	0.01	0.45	142.08
mRMR 05	16.29	0.02	0.01	0.01	0.48	146.50
mRMR 10	12.43	0.02	0.01	0.01	0.53	142.84
mRMR 20	7.72	0.01	0.02	0.01	0.58	198.56
mRMR 30	5.46	0.02	0.03	0.01	0.49	86.29
mRMR 40	6.26	0.05	0.04	0.01	0.52	102.73

Based on the results presented in Table 2, the KNN and RF classifiers and, mainly, SVM have a higher training time than the other approaches. Nevertheless, the application of the mRMR selection technique (with 5 and 10 characteristics) reduces the training time of the RF classifier, enabling its deployment for SEs, achieving a time closer to DT classifier.

Similarly to the training time, the detection time (presented in Table 3) of the KNN and SVM classifiers are longer than the other ML techniques. However, differently from the training time, the impact of the selection techniques are lower. In general, the LR, NB and DT techniques spend very small time performing the detection. Close to them is the RF, proving to be a feasible solution too.

4.2.4 Volume of Data

Regarding the volume of data transmission, the raw data is in the PCAP format (produced by the network monitoring), which is a complex format that generates a high volume of data. For example, in 5 minutes of monitoring, almost 60MB of raw data should be transmitted from the Fog to the Cloud.

Therefore, the network infrastructure suffers an unnecessary overhead. For instance, a 20Mbps network spends almost 2 minutes transmitting the data from the Fog to the Cloud.

In the proposed system, after the feature extraction, 80 features are created and later these features are selected by a specific technique. Thus, the data processing flow tends to reduce the volume of data to be transmitted from the Fog to the Cloud. Table 3 shows the results of the volume of data generated in each stage of the data processing flow using the dataset described in Section 4.1.

Table 3. Volume of Raw Data and Processed Data

Approach	Volume of Data
Raw Data	15.16Gb
Extraction (80 Features)	4.17Gb
LV (63 Features)	88Mb
SVC (18 Features)	21Mb
ET (22 Features)	22Mb
Lasso (64 Features)	84Mb
mRMR (5 Features)	5Mb
mRMR (10 Features)	9Mb
mRMR (20 Features)	22Mb
mRMR (30 Features)	37Mb
mRMR (40 Features)	50Mb

When feature selection occurs in the Fog, the amount of data reduces from 15.16GB (raw) to 25MB (processed), representing less than 0.2% of the volume of data. Thus, the Fog-Cloud integration approach increases network scalability, while it causes a very low impact in network resources availability.

4.3. Final Discussion

The results of the experiments highlight the importance of the features selection for the accuracy, execution time and volume of data. For example, using the most appropriate selection technique, the performance of the KNN and SVM classifiers increases by 8% and 7%, respectively. Additionally, the LR technique using the 80 extracted features (no selection) has an unacceptable accuracy, while using the Extra-Tree technique, it achieves more than 93% of accuracy.

Regarding the training time, its importance increases in contexts that a recurrent training is necessary to update the ML model to the high dynamics of the SE, such as smart campi and smart cities. Thus, the ML model will be trained in a very short time period to keep the detection of DDoS attacks effectively. The same reasoning can be applied to the detection time. In such a context, the DT and RF with mRMR-10, mRMR-20, Lasso or SVC are the suitable combinations, since they are fast, have a high accuracy and generate small volumes of data. On the other hand, if the periodicity of the training is longer, due to static behavior of the SE (such as a Smart Industry), other approaches are feasible.

5. Conclusion and Future Work

Nowadays, new paradigms have emerged, such as Smart Environments (heterogeneous IoT and Personal devices). A critical challenge in smart environments lies in the detection of network DDoS attacks, resulting from security vulnerabilities. Their early detection helps to avoid the QoS degradation and possible financial losses.

This article described an Intelligent System for detecting DDoS in ESs. The proposed system is based on ML techniques to perform the traffic segmentation and DDoS detection, while a features selection approach is applied to reduce the amount of data exchanged between Fog and Cloud and to improve the accuracy of the detection.

Results from performance evaluation based on real traffic as workload indicate a 99% of accuracy (in average) to detect DDoS attacks, while the training time was 10 seconds (in average). As future work, we intend to investigate new security solutions for other threats to SEs, such as Side-Channel, OS Service Scan, Keylogging and Data Exfiltration.

References

[1] Abdulkareem, K.H., Mohammed, M.A., Gunasekaran, S.S., Al-Mhiqani, M.N., Mutlag, A.A., Mostafa, S.A., Ali, N.S., Ibrahim, D.A.: A review of fog computing and machine learning: Concepts, applications, challenges, and open issues. IEEE Access 7, 123–140 (2019).

[2] Ahmed, E., Yaqoob, I., Gani, A., Imran, M., Guizani, M.: Internet-of-things-based smart environments: state of the art, taxonomy, and open research challenges. IEEE Wireless Communications 23 (5), 10–16 (2016).

[3] Andrea, I., Chrysostomou, C., Hadjichristofi, G.: Internet of things: Security vulnerabilities and challenges. In: 2015 IEEE Symposium on Computers and Communication (ISCC), pp. 180–187 (2015).

[4] Brun, O., Yin, Y., Augusto-Gonzalez, J., Ramos, M., Gelenbe, E.: Iot attack detection with deep learning. In: ISCIS Security Workshop (2018).

[5] Chang, C.C., Lin, C.J.: Libsvm: A library for support vector machines. ACM transactions on intelligent systems and technology (TIST) 2 (3), 1–27 (2011).

[6] Diro, A.A., Chilamkurti, N.: Distributed attack detection scheme using deep learning approach for internet of things. Future Generation Computer Systems 82, 761–768 (2018).

[7] Doshi, R., Apthorpe, N., Feamster, N.: Machine learning ddos detection for consumer internet of things devices. In: 2018 IEEE Security and Privacy Workshops (SPW), pp. 29- 35. IEEE (2018).

[8] Firouzi, F., Farahani, B.: Architecting IoT Cloud, pp. 173-241. Springer International Publishing, Cham (2020).

[9] Friedman, J., Hastie, T., Tibshirani, R.: Regularization paths for generalized linear models via coordinate descent. Journal of statistical software 33 (1), 1 (2010).

[10] Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine learning 63 (1), 3–42 (2006).

[11] HaddadPajouh, H., Dehghantanha, A., Khayami, R., Choo, K.K.R.: A deep recurrent neural network based approach for internet of things malware threat hunting. Future Generation Computer Systems 85, 88 – 96 (2018). DOI: https://doi.org/10.1016/j.future.2018.03.007

[12] Hamamoto, A.H., Carvalho, L.F., Sampaio, L.D.H., Abrão, T., Proença Jr, M.L.: Network anomaly detection system using genetic algorithm and fuzzy logic. Expert Systems with Applications 92, 390–402 (2018).

[13] Kaushik, S.: Introduction to feature selection methods with an example (or how to select the right variables?). Analytics Vidhya (2016).

[14] Koroniotis, N., Moustafa, N., Sitnikova, E., Turnbull, B.: Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. CoRRabs/1811.00701(2018). URL http://arxiv.org/abs/1811.00701

[15] Li, H., Ota, K., Dong, M.: Learning iot in edge: deep learning for the internet of things with edge computing. IEEE Network 32 (1), 96–101 (2018).

[16] Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., Elovici, Y.: N-baiot—network-based detection of iot botnet attacks using deep autoencoders. IEEE Pervasive Computing 17 (3), 12–22 (2018).

[17] Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., Elovici, Y.: N-baiot—

network-based detection of iot botnet attacks using deep autoencoders IEEE Pervasive Computing 17 (3), 12–22 (2018).

[18] Peng, H., Long, F., Ding, C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on pattern analysis and machine intelligence 27 (8), 1226–1238 (2005).

[19] Pisani, F., de Oliveira, F.M.C., Gama, E.S., Immich, R., Bittencourt, L.F., Borin, E.: Fog computing on constrained devices: Paving the way for the future iot (2020).

[20] Sharafaldin, I., Lashkari, A.H., Hakak, S., Ghorbani, A.A.: Developing realistic distributed denial of service (ddos) attack dataset and taxonomy. In: 2019 International Carnaha Conference on Security Technology (ICCST), pp. 1–8. IEEE (2019.)

[21] Sivanathan, A., Gharakheili, H.H., Loi, F., Radford, A., Wijenayake, C., Vishwanath, A., Sivaraman, V.: Classifying iot devices in smart environments using network traffic characteristics. IEEE Transactions on Mobile Computing 18 (8), 1745–1759 (2018).

[22] Vinayakumar, R., Alazab, M., Srinivasan, S., Pham, Q.V., Padannayil, S.K., Simran, K.: A visualized botnet detection system based deep learning for the internet of things networks of smart cities. IEEE Transactions on Industry Appli cations (2020).

[23] Vishwakarma, R., Jain, A.K.: A survey of ddos attacking techniques and defence mechanisms in the iot network. Telecommunication Systems 73 (1), 3–25 (2020).

[24] Yamauchi, M., Ohsita, Y., Murata, M., Ueda, K., Kato, Y.: Anomaly detection for smart home based on user behavior. In: 2019 IEEE International Conference on Consumer Electronics (ICCE), pp. 1–6. IEEE (2019)

[25] Zhou, Y., Cheng, G.: An efficient network intrusion dete tion system based on feature selection and ensemble classifier. CoRR abs/1904.01352 (2019).

[26] Zagrouba, Rachid, & Reem AlHajri. "Machine Learning based Attacks Detection and Countermeasures in IoT." International Journal of Communication Networks and Information Security (IJCNIS) [Online], 13.2 (2021): n. pag. Web. 30 Nov. 2021.