Intelligent VNF Placement to Mitigate DDoS Attacks on Industrial IoT

Guilherme Werneck de Oliveira, **Michele Nogueira**, **Aldri Luiz dos Santos** and **Daniel Macêdo Batista**

***Abstract*—The Internet of Things (IoT) has undergone rapid popularization, reaching a wide range of application domains, such as manufactures. Hence, more and more heterogeneous IoT devices have been deployed in a variety of industrial environments, progressively becoming common objects to the supply chain. The physical infrastructure of manufacturing systems has become complex and requires efficient and dynamic solutions for managing network performance and security. Network Function Virtualization (NFV) has attracted attention when the intention is to respond to security threats on Industrial IoT (IIoT). Few works use NFV to detect and mitigate security threats on IIoT networks, but even less consider performance indicators of the network context when placing the Virtual Network Functions (VNFs). Thus, this work introduces a Machine Learning (ML) approach to place security VNFs based on NFV performance to mitigate Distributed Denial of Service (DDoS) attacks on IIoT. Experiments considering a new composed data set and diverse ML techniques show ML classification as an alternative for IIoT scenarios, achieving, according to the best-performing technique, 99.40% of accuracy in relation to the ideal placement. To facilitate the reproduction of the work, all the code and data produced are publicly available.**

***Index Terms*—Industrial IoT, NFV, VNF Placement, Performance Management, DDoS Attacks, Machine Learning.**

I. Introduction

THE Internet of Things (IoT) has undergone rapid popularization, reaching a wide range of application domains, such as healthcare, environmental monitoring, home automation, smart mobility and Industry 4.0 [1], [2], [3]. Thus, more and more smart devices are deployed in a variety of public and private environments, progressively becoming common objects of everyday life. IoT has been seen by many as the next stage of the Internet by enabling evolution in society and industry, like smart manufactures, because with IoT, supply chain, energy systems and other systems, become more efficient.

The definition of IoT related to industrial infrastructure is referred as Industrial IoT (IIoT), in which many heterogeneous smart devices are deployed to optimize production methods [4]. The physical infrastructure of heterogeneous systems, such as smart manufacture, is complex and requires efficient and dynamic solutions for managing, protecting and configuring networks [5], [6].

Guilherme Werneck de Oliveira and Daniel Macêdo Batista are with the Computer Science Department, University of São Paulo, Brazil. E-mails: werneck.guilherme@alumni.usp.br, batista@ime.usp.br.

Michele Nogueira and Aldri Luiz dos Santos are with the Computer Science Department, Federal University of Minas Gerais, Brazil. E-mails: {michele, aldri}@dcc.ufmg.br.

Two requirements that stand out on IoT are commonly performance and security, mainly when applying it on manufactures. However, many devices on IoT do not support strong security mechanisms and therefore can be target and even means for a series of attacks [7]. Among various types of attacks, the severity of Distributed Denial of Service (DDoS) attacks affects both networks formed by IoT devices and the Internet in general [8]. According to the company Kaspersky [9], attacks on IoT devices have become frequent. In the first six months of 2019, there were 105 million attacks on IoT devices from 276,000 unique IP addresses. These attacks take advantage of several vulnerabilities, such as one discovered in August 2020 by the company Check Point [10] against Alexa personal assistant, which allowed the manipulation of its tokens and the carrying out actions on behalf of victims. Another important finding is that 28% of companies using IoT devices report having encountered incidents involving connected devices. With large volume of data generated by sensors and smart devices, including industrial sensitive data, the results of such incidents can be severe.

One approach that has been gaining ground when responding to IoT threats is the use of virtualized network resources through Network Function Virtualization (NFV) [5], [11]. NFV introduces a new degree of flexibility and scalability by creating on-demand virtual network capabilities such as firewalls, Intrusion Detection Systems (IDS), and Deep Packet Inspection (DPI) systems. Virtualization allows multiple instances of a specific detection or mitigation mechanism to be deployed at different locations on the network, given restrictions imposed by malicious events [11] and the limited processing and storage power of IoT devices.

One of the main concerns related to NFV lies in the performance management of virtualized resources. For [12], the real challenge is producing efficient and scalable software to orchestrate virtual networks to allow these networks to be easily configured and have their lifecycles managed. In [13], in addition to resource management and orchestration, the authors also consider energy efficiency, the efficiency of the provisioned resource itself, modeling and resource allocation as open challenges. All this is related to performance management in virtualized environments, a topic that has raised several research challenges, such as Virtualized Network Functions (VNF) orchestration, Service Function Chaining (SFC) optimization, among others [14], [15].

A challenge highlighted in this work is VNFs placement for DDoS attacks. Existing NFV-based mechanisms against threats on IoT networks ignore the application and context. There is no solution that use NFV and network perfor-

mance indicators as basis to DDoS attack mitigation through VNF placement. A previous work [16] presented results for preliminary tests to determine the effectiveness of different Machine Learning (ML) methods to predict the response time of IIoT applications. This previous solution has considered applications composed by microservices along with network and fog servers performance indicators. Results obtained from these preliminary tests have shown that network environment information, such as latency and capacity, as well as service processing information, are essential for decision making when the objective is to optimize the Quality of Service (QoS) provided to end users.

This article introduces an ML approach to place security VNFs on fog servers aiming an efficient mitigation of DDoS attacks on IIoT environments. The approach differs from similar ones by considering NFV performance indicators, such as the deployment time and CPU/RAM consumption. The approach is based on ML classification techniques, that can be customized. Simulation experiments evaluate the performance of the approach considering different supervised and unsupervised ML techniques when allocating security VNFs near the attacked IIoT devices and near the attacker device. Supervised techniques have achieved the best results when the VNFs were allocated near the attacker device. For instance, the accuracy obtained with the XGBoost technique was 99.40%.

The main contributions of this article are two-fold: i) a proposal and analysis of supervised and unsupervised ML models for automated security VNF placement, by using classical classification techniques and considering different performance indicators; ii) organization and discussion of existing approaches that applied NFV to detect and mitigate IoT attacks considering different performance indicators. As a secondary contribution, this article develops, and makes publicly available, a new data set describing the execution of IIoT applications in a fog environment in face of attacks. The new data set is founded on the Fog IIoT Factory data set [16] and on the BoT-IoT data set [17]. Moreover, all code and data produced in this article are publicly available, allowing the reproduction of the experiments by the community.

This article proceeds as follows. Section II introduces the background about performance and security with NFV. Section III reports related work and an analysis about them. Section IV details the ML approach with the composed data set, the processes and techniques used. Section V presents results obtained and discussions about them. Finally, Section VI concludes the article.

II. Background

Performance indicators are directly related to resource metrics that demonstrate the degree of functionality of a system and, therefore, represent its state in an abstract way. From an NFV point of view, the European Telecommunications Standards Institute (ETSI) [18], [19] defines performance metrics related to speed, accuracy and reliability according to categories referring to orchestration, virtual machine operation, establishment and operation of networks and technology components such as services. These metrics were defined for entities that provide VNFs and that manage virtualized infrastructures, in order to guarantee that offered resources have the necessary performance according to the predicted quality requirements. For example, a network that is being highly demanded by benign requests, that are not attacks, and has a high virtual machine provisioning latency for the virtualized firewall service, can generate large delays in responses to requests, which directly affects the end-user QoS.

In the same vein as ETSI, the Internet Engineering Task Force (IETF) [20] designated a specific group, called Benchmarking Methodology Working Group (BMWG), to produce a series of recommendations on key characteristics and performance analysis of network devices, systems and services. As a result, the Benchmarking Methodology for Network Virtualization Performance was developed, considering as performance metrics: CPU and RAM consumption, data transfer rate, packet frame loss rate and network latency.

There are metrics related to NFV in other works. In [14], for example, the authors address issues related to NFV fault tolerance and present metrics specific to the problem, such as: packet loss, average packet throughput, congestion interconnection, among others. The work of [21] presents performance comparison metrics for SDN and NFV controllers. NFV metrics focus on vCPU and vMemory (compute-associated), latency and throughput (communication-associated), I/O rate, and recovery time of VNFs (storage-associated). In [15], the authors use metrics such as CPU usage, packet throughput, and latency to present how the effect of Non-Uniform Memory Access (NUMA) and service positioning impact NFV performance.

The characteristics of IoT devices are considered on network implementation, such as low energy consumption and limited computing resources. One should mainly observe the communication behavior between devices and services available to them. RFC 8172 [22] adds other metrics to NFV performance, such as time to deploy and migrate VNFs, which must be taken into account so that response time on IoT network is not adversely affected.

NFV applied specifically to security on IoT is attracting attention [5], [11], [23]. Security functionality provided by VNF itself may have stricter requirements, especially in terms of response time, to ensure that the detection and mitigation of an attack is carried out before it is too late. For example, an IDS installed on the local IIoT network can act within the expected time due to the fact that it is located in the exact place where the flows of interest are traveling. Forwarding this traffic to a remote computing environment, even if that environment has high processing capacity, can make IDS useless if latency between local network and remote environment is too high.

Metrics and measurements are basis for the monitoring process of any type of network, being used to determine the components state and, also, to serve as inputs for performance management. In conventional networks, performance considers issues such as data transfer, network response time, packet loss, percentage of resource usage, data connection utilization, and so on. However, when a network context includes IoT devices together with NFV-based security approaches, other issues raise and guide the analysis of their behavior and

the impact they have on performance. Examples are on **IoT domain** – types of devices and communication protocols used by them; in **NFV** – provisioning, scaling, network connections and orchestration of VNFs; and in **security** – tools, algorithms, detection time and threat mitigation.

When fully operational, a security mechanism should aim at two main actions: detection and mitigation of threats to the network. Threats detection can be considered a preliminary step in the security incident response process. Likewise, mitigation can be seen as a subsequent step, which objective is to neutralize the threat present on network. For example, once a DDoS attack is identified by a detection engine, the mitigation system can take action in order to deny or redirect malicious packets for disposal. Understanding and evaluating these mechanisms help to understand what level of completeness and maturity of NFV the solution is at.

ML techniques can assist the mitigation of security threats on IoT and, more specifically, on IIoT. For instance, to classify the traffic and identify suspect anomalies as soon as possible, to predict the performance of the mitigation mechanisms before they are instantiated in a virtual environment, and to find the best location to instantiate these mitigation mechanisms through VNFs as proposed in this article.

III. Related Work

Recently, several studies have investigated how flexibility and scalability of NFV can be exploited to respond to IoT threats. In [24], the authors propose a framework which objective is to manage and orchestrate security policies on IoT networks. Heterogeneity of physical and virtual devices prevails. The framework architecture is composed of user, orchestration, security application and management planes. To demonstrate the framework applicability, the authors describe two case studies and test their performance in a proof of concept in [23]. The first case study considers a Multi-access Edge Computing (MEC) scenario, which the main objective is to support the ability to provide VNFs closer to IoT devices, ensuring that network quality requirements are met. The second case study presents a scenario related to smart buildings, Building Management System (BMS), in which protection mechanisms to different devices, such as computers and air conditioners, can be recommended by security policies.

The proof of concept was focused on applying policies to provision security mechanisms on Software-Defined Networking (SDN). The experiments considered the comparison between three approaches. The first and second were implemented by OpenDaylight (ODL) and Open Network Operating System (ONOS) SDN controllers, respectively, and the third approach by a firewall VNF component based on Network Configuration Protocol (NETCONF) and iptables. They also considered an environment under Man-in-the-middle (MITM) attack already detected, using the isolation of compromised sensors as a mitigation strategy. On the IoT devices, a virtual machine was used running a Cooja Contiki emulator, contained by IoT sensors connected through the 6LoWPAN protocol. Compared to ODL and ONOS SDN controllers, the firewall implementation in VNF had a lower performance, especially in the policy enforcement stage. According to the authors, the type of communication was the main reason for this performance loss.

In [25], the authors present an approach to manage Authentication, Authorization and Accountability (AAA) policies on IoT through a VNF called vAAA. It is established at fog level along with security policies, which run on a cloud level SDN controller. The authors also propose to carry out communication between IoT devices and network services through protected virtual channels (vChannel-Protection), in particular, using DTLS. For experimentation, IoT devices were used on a custom version of Contiki OS 2.7 and the erbium CoAP server, along with 6LoWPAN protocol. The proposed components performance was evaluated according to the secure communication process between IoT devices and the broker, which consists of the authentication, authorization and protected channel steps from refinement, translation and security policy enforcement actions. Values presented for authentication, authorization and protected channel steps remained constant for all actions, except for the policy application in the protected channel step.

In [26], the authors present an architecture regarding security management for the framework proposed initially by [24]. At this stage, the authors developed a management module composed by a component called data filtering and pre-processing, which captures information from the network and forwards it to the incident detector. The latter performs the security analysis, looking for possible attacks on the network. The module also contains a database of attack signatures and another component based on artificial intelligence, which applies ML techniques to detect behavioral anomalies, a component developed by [27]. For detection and mitigation, that is, the application of security policies on the network, the average time between translation and application was close to each other.

Following the framework [24] evolution, the work of [28] brings a new approach to mitigating attacks on IoT network through the automated implementation of honeynets based on VNFs. The approach is about the creation of network virtualized IoT devices (vIoTHoneynet) without any function so that, upon detection of an attack, traffic is redirected to vIoTHoneynet through updated rules in SDN controller. The experiments were based on two BMS scenarios. Considering that vIoTHoneynet is instantiated on demand, the experiments considered two types of times to compose the total mitigation time. The first time was related to the vIoTHoneynet establishment, for example due to the compilation of the code, and the second time was related to the vIoTHoneynet configuration on the network, for example, by configuring the routing of the attacker's connections through the implementation of new rules in SDN controller and by the application of the security policies. For the second scenario, the average setup time was three times bigger than the first one. When the vIoTHoneynets configuration times were added to the security policy enforcement time to mitigate the attack, the same behavior was observed for the scenarios.

Regarding ML application for anomaly detection on IoT networks, the work of [27] proposes a model based on

supervised learning developed and proposed by [24]. The authors implement a virtual intelligence agent that receives information from the network monitoring module and analyzes it in order to detect patterns referring to attacks such as DDoS, Probing, User to Root (U2R) and Remote to Local (R2L). The Random Forest, J48, Bayesian Network and Hoeffding Tree learning models were developed and trained from data sets that include the aforementioned attack types. The experiments were based on collecting data from temperature and CO2 sensors from different rooms. For both kind of sensors, the models showed a satisfactory accuracy for all types of attacks.

In the same ML context, [29] presents a solution called NETRA, which provides VNFs on IoT network edge in order to improve its security. It is based on Docker containers to support security VNFs. The network core layer is build up from the Internet Service Provider connection and the Docker Hub, responsible for storing images used in each VNF. The edge layer contains the IoT gateway developed on Raspberry Pi 3, in which the VNFs for that security domain are implemented. The IoT devices layer is composed by the smart devices used on the network. All network traffic performed at the edge is analyzed by the vAnalytics component, which implements a ML model based on Random Forest to classify attacks based on identified anomalies. The experiments were carried out under two implementation perspectives, virtual machines and containers. The authors presented an evaluation of the learning model performance for DDoS detection and mitigation attacks. In this case, the model presented an accuracy of 94.40% and an average VNF response time for mitigation of less than one second.

[30] introduced a container-based approach to edge computing architectures. The authors conducted a performance comparison in which the Suricata IDS was run on a Raspberry Pi 3 device, which includes two different scenarios: the first on a physical machine (SoBM) and the second, on a Docker container (SoDC). The performance indicators used in comparison were: number of processed packets, transfer rate, RAM and CPU consumption and number of dropped packets. In relation to the number of processed packets, transfer rate and RAM memory consumption, the values were very close, being able to consider a tie between two implementations. However, the CPU consumption and number of dropped packets indicators pointed out that the SoDC approach has an advantage over SoBM, which the CPU consumption and the number of dropped packets was lower.

In [31], the authors use scalability advantages provided by NFV to implement an IDS based on ML models to mitigate malwares on IoT networks. The solution presented is composed of an RNN-LSTM model, executed in a centralized network, which detects attacks on the network, and creates surveillance zones through different NFVs to carry out the application of security patches on the devices. The authors performed experiments considering SEIR epidemic model, used in the epidemiology domain. In this model, there are susceptible (S), exposed (E), infected (I) and resistant (R) individuals. In this case, individuals are IoT devices. From the malware dissemination on the IoT network, the performance of the approach was measured according to the network scalability, i.e., the creation of surveillance zones in NFV and the amount of viruses on the network. The results showed that the creation of virtualized surveillance zones remains constant for 1000 devices on the network. In this case five zones were created. The increase in the number of surveillance zones, to 50, started from 10,000 devices on the network. However, it remained constant until 100,000.

Based on fog computing, [32] proposes a VNF implementation to perform DDoS attacks detection on IIoT networks. The anomaly detection approach proposed by the authors considers the Snort IDS usage to implement state machines modeled from TCP and proprietary communication protocols used in industrial control systems, in this case, Modbus. The results presented showed that an architecture that considers the detection system process close to IoT devices, on fog level, is more advantageous in relation to both evaluated indicators, average response time and detection rate.

In a context of smart home solutions, [33] present a proposal for SDN controllers usage to reinforce static and dynamic access control to the IoT network. In addition, they present the implementation of a VNF to mitigate Address Resolution Protocol (ARP) spoofing attacks. The ARP server implementation was carried out in two ways: one using the Scapy Python library (SC) and the other using the Data Plane Development Kit (DPDK). This server aims to be a reliable entity that manages ARP requests on the network. In this way, all ARP packets on the network are filtered by a SDN controller and forwarded to the server for resolution. To demonstrate the mitigation approach, experiments were performed using Mininet, according to different amounts of machines present on the network. The VNF response time implemented in SC was more than 70 milliseconds for ten devices, and approximately one second for 50 devices. For the DPDK implementation, the average response time was 0.57 millisecond.

After an extensive literature review, it was possible to elaborate the Table II that provides an overview of performance indicators for security on IoT context. To support related work analysis, Table I points out each category of metrics used in the performance evaluation and the amount of works in the literature. The most used attack to demonstrate the NFV usage is DDoS (55.55%), ahead MITM (33.33%) and others (33.33%). In addition, it is also possible to observe that 46.15% of works consider NFV for attacks prevention on IoT networks, 53.85% for detection and 69.23% for mitigation. For IoT context, 46.15% of works address the characteristics of fog computing for the NFV application. However, only [32] presents a comparison between detection and mitigation performance for different architectures.

Metrics related to components implemented through virtualization (NFV) are present in 50% of works. Metrics related to the network (Network) are in just 20%. The ones that stand out the most are the metrics that show effectiveness of the proposed solution (Solution), that are in 80% of the works. NFV and Network performance indicators are essential to compose security solutions that aim to increase the network QoS level. However, they are less explored in approaches found in the literature than those used to demonstrate the

TABLE I: Categories, performance metrics and number of works.

Category	Metric	Number of works (non-excludable)
NFV	CPU consumption	5
	RAM consumption	
	Storage	
	Scalability	
	Environment deployment time	
Network	Latency	2
	Throughput	
	Number of processed/discarded packages	
Solution	Attack response time	8
	Security policy enforcement time	
	Attack detection and mitigation time	
	ML model accuracy	

solution effectiveness. This characteristic has a negative impact on the management of virtualized resources, as it is not enough for the approach to have high problem resolution rates. It is also necessary to consider variables of the environment in which the solution is executed, or the context performance.

Table III compares all the related works with the one proposed in this article. The proposed approach here is the only one aiming to place security VNFs that mitigate DDoS attacks on the IIoT by composing performance indicators of NFV, network and solution.

IV. ML Approach for VNF Placement

Performance management assumes how to use indicators or metrics collected from network to efficiently provision resources necessary to maintain the QoS delivered to its users [34], [35]. For the operation of an NFV be carried out efficiently, performance indicators must be well defined and measured to demonstrate the real demand, the behavior and the health of a network. The prediction of network behavior brings advantages for many applications that have strict QoS requirements by anticipating the placement of devices and resource allocation, besides allowing service designers to predict the performance of their applications [16], [36].

Thus, NFV performance indicators can contribute to ML decision-making on whether or not to use a particular virtualized resource for an identified purpose in specific locations. In other words, NFV performance management can be supported by ML decisions. For instance, for VNF placement mechanisms presented in [37], [38], [39], labeling, training and classifying specific servers to deploy a VNF can be helpful instead of just considering the problem's solution by optimization techniques, such as time-slot decoupled, randomized rounding approximation, heuristic, graph partitioning, greedy and game theory algorithms, which may consume long execution time and high computational power.

Algorithm 1 describes the proposal of this work. It is based on decisions of previously trained ML models to find the best server on which a security VNF can be deployed to mitigate DDoS attacks by composing different performance indicators. To test the effectiveness of this approach, we evaluate the performance of supervised and unsupervised ML models, which are presented on Subsection IV-A.

Algorithm 1 Security VNF Placement.

Require: X, Y, m ▷ X is the training set of IIoT network scenarios, Y is the testing set of IIoT network scenarios and m is a ML model
while $X \neq \emptyset$ **do** ▷ training step
$x \leftarrow X$ ▷ x is an IIoT network scenario of X
$x\_train \leftarrow get\_train(x)$ ▷ x_train is the set of IIoT performance indicators of x
$x\_target \leftarrow get\_target(x)$ ▷ x_target is the target fog server where the security VNF is placed of x
$train\_model(m, x\_train, x\_target)$ ▷ training the model according to performance indicators and the target fog server
end while
while $Y \neq \emptyset$ **do** ▷ testing step
$y \leftarrow Y$ ▷ y is an IIoT network scenario of Y
$y\_pred \leftarrow get\_train(y)$ ▷ y_pred is the set of IIoT performance indicators of y
$pred\_fog\_server \leftarrow predict(m, y\_pred)$ ▷ predicting the target fog server according to trained model m
$Fog\_Servers \leftarrow pred\_fog\_server$ ▷ Fog_Servers is the set of predicted fog servers where security VNFs are placed
end while
return $Fog\_Servers$

A. ML Models For Security VNF Placement

Conducting an analysis of both learning methods, supervised and unsupervised, helps us to deeply understand how the IoT networks context influences decision making. However, we cannot limit the models evaluation to just these methods. It is also necessary to evaluate how the algorithms established in the literature behave when the defined performance indicators feed them. For supervised learning classification, we implemented and evaluated Decision Tree, Random Forest, Extreme Gradient Boosting (XGBoost), Logistic Regression, Support Vector Machine, K-Nearest Neighbors, and Naive Bayes. For unsupervised learning, we implemented and evaluated K-Means, Hierarchical Cluster, and Gaussian Mixture Model.

Decision Tree classifier is a binary tree in which predictions are made by going through from root to leaf, where is associated a class, in our case, a specific fog server to deploy a security VNF. It implements Gini impurity concept to describe how homogeneous a node is. A node is homogeneous if all its samples belong to the same fog server, while a node with many samples from many different fog servers is less pure. Unlike Decision Tree, Random Forest creates small decision trees with several random sets of features, or the performance indicators. After this creation, predictions are made from votes. Each small tree makes a decision based on the data presented and, the most voted decision, is the final parameterized algorithm. XGBoost is an implementation

TABLE II: Performance indicators overview for security and IoT context.

Work	Evaluated performance	Security		IoT		
		Approach	Attack	Protocol	Architecture	Device
[33]	ARP response time	Prevention and mitigation	MITM	Not stated	Not stated	General
[23]	Security policy enforcement time	Prevention and mitigation	MITM	6LoWPAN	Not stated	General
[30]	Number of processed packets, number of dropped packets, transfer rate, CPU and RAM consumption	Detection and mitigation	Not stated	Not stated	Fog	Raspberry Pi 3
[29]	Storage, RAM consumption, latency, throughput, scalability, environment deployment time and ML model accuracy	Detection and mitigation	DDoS	Not stated	Fog	Raspberry Pi 3 and IP cameras
[26]	Time to detect and enforce security policies	Detection and mitigation	DDoS and malware	CoAP and 6LoWPAN	Fog	General
[25]	Security policy enforcement time, CPU and RAM consumption	Prevention and mitigation	MITM	CoAP and 6LoWPAN	Fog	General
[32]	Detection time	Detection and mitigation	DDoS	TCP and Modbus	Fog	General
[27]	ML model accuracy	Detection	DDoS, Probing Attack, U2R and R2L	Not stated	Not stated	Temperature and CO2 sensors
[31]	Scalability	Detection and mitigation	Malware	Not stated	Not stated	General
[28]	Time to deploy the environment and apply security policies	Detection and mitigation	DDoS	CoAP, RPL and LoWPAN	Not stated	Humidity, temperature, light, CO2 and RFID sensors
This work	CPU and RAM consumption, environment deployment time, latency, throughput, network response time, number and size of processed packets, packet rates, application ports, approach response time and ML model accuracy	Detection and mitigation	DDoS	Not stated	Fog	IIoT sensors

TABLE III: Comparison of related works with this work.

Work	Performance indicators			Performance indicators as decision variables
	NFV	Network	Solution	
[33]	No	No	Yes	No
[23]	No	No	Yes	No
[30]	Yes	Yes	No	No
[29]	Yes	Yes	Yes	No
[26]	No	No	Yes	No
[25]	Yes	No	Yes	No
[32]	No	No	Yes	No
[27]	No	No	Yes	No
[31]	Yes	No	No	No
[28]	Yes	No	Yes	No
This work	Yes	Yes	Yes	Yes

of Gradient Boosted Decision Trees designed for speed and performance. It aims to minimize the loss function, choosing distinct optimal adjustment values for each region of the tree instead of a single one for the entire tree. Based on this concept, XGBoost calculates the fit of its sequential trees according to the weighted errors of predecessor trees to predict the fog server that is target of the model.

Beyond tree-based structures, logistic regression estimates the probability associated with the occurrence of a given event, the security VNF placing, in face of a set of explanatory variables, the performance indicators. Its coefficients are estimated from the data set, by the maximum likelihood method, which finds a combination of coefficients that maximizes the probability of the fog server having been observed. The Support Vector Machine objective is to find a hyperplane in an n-dimensional space, where n is the number of performance indicators, that distinctly classifies the fog server as target. Although logistic regression and Support Vector Machine are intended for classifying binary variables, we tested its behavior for multi fog servers in order to look for some possible pattern in the performance indicators composition.

K-Nearest Neighbors classifies a new fog server according to the identification of its k nearest neighbors, that is, it searches for groups of fog servers in common based on their similarities, the performance indicators that describe the VNF placement decision. It is a lazy algorithm as it memorizes the training data set and classifies the new fog server only if there is numerical proximity between the learned data. The basic reasoning of Naive Bayes is to disregard the correlation among performance indicators, as it treats each one independently, or naively. With this algorithm, we were able to observe how the high or low performance indicators correlation of the IIoT

environment defined here impacts the model classification.

Regarding unsupervised learning classification, K-Means evaluates and clusters fog servers according to their characteristics. From the k definition, or the predicted number of fog servers, the algorithm defines centroids in the plane and separates the fog servers in k groups of equal variance. The Hierarchical Cluster aim is to produce a hierarchical series of nested clusters. It begins assuming each fog server as a separate cluster and then identifies two clusters which can be closest together and, finally, merge two maximum comparable clusters. This hierarchy of clusters is represented as a tree in which the root is a unique cluster that gathers all the samples and leaves being the clusters with only one sample. As a probabilistic model, Gaussian Mixture Model considers the fog servers are generated from a mixture of a finite number of Gaussian distributions with unknown parameters, or the performance indicators.

Another important perspective to analyze is the importance of performance indicators for the models mentioned. This analysis demonstrates the importance score for each feature, depicting the importance of that feature for the security VNF placement. And, in the context presented here, it aims to show the proportional contribution of performance indicators that make up the learning of models.

V. Performance Evaluation

We present an experimental comparison of ML models implemented using Scikit-learn for all scenarios composed in the new data set. The Jupyter Notebooks used in the experiments are publicly available on Github[1]. We evaluated three types of scenarios: *small* (10 IIoT and 10 fog nodes), *medium* (25 IIoT and 15 fog nodes) and *large* (50 IIoT and 25 fog nodes). In these scenarios, applications are executed on IIoT nodes, which requires instantiation of some microservices on the fog nodes. Fig. 1 summarizes these scenarios. In some moments, these IIoT nodes may suffer DDoS attacks and the ML techniques will choose some fog node to instantiate a security VNF, which could be a DNS sinkhole, a DPI or a firewall for instance. The selection of the best fog node will consider the performance indicators in Table V. The universe of data used for training the models contains 84 network scenarios, accounting 880 nodes for small (44*10 IIoT + 44*10 fog nodes), 1000 nodes for medium (25*25 IIoT + 25*15 fog nodes) and 1125 nodes for large (15*50 IIoT + 15*25 fog nodes) topology. To test the proposed models, the data universe includes 47 scenarios that total 880 nodes for small (44*10 IIoT + 44*10 fog nodes) and 240 nodes for large topology with 30 fog nodes instead of 25 (3*50 IIoT + 3*30 fog nodes). The difference in the amount of fog nodes is due to the original Fog IIoT Factory data set and was kept to also evaluate models performance in distinct structures than those used for training.

Accuracy (1), precision (2), recall (3) and F1-score (4) metrics were used to evaluate models performance. On the IIoT network context stated, the correct security VNF placement on a specific fog server is equivalent to true positive (TP), an

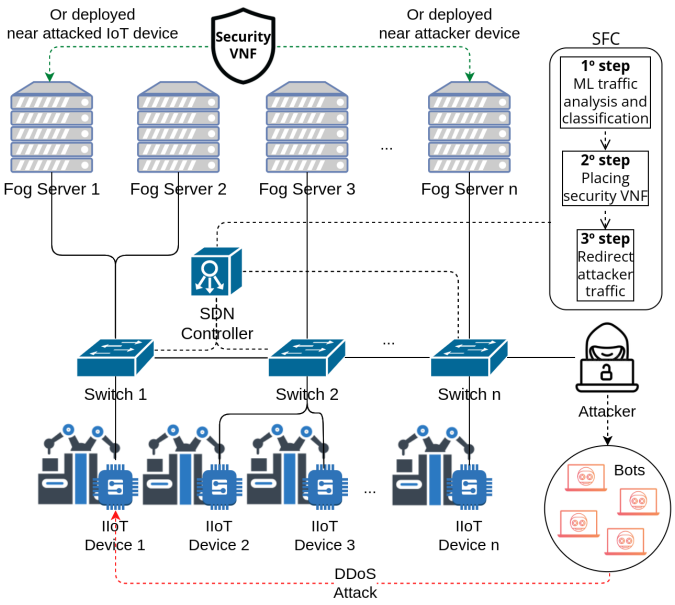


Fig. 1: Deploying security VNF on IIoT network under DDoS attacks.

outcome in which the model correctly predicts the positive class, and true negative (TN), when the model correctly predicts the negative class. False positive (FP) represents the security VNF placement even when the network scenario do not suffer DDoS attacks. Likewise, false negative (FN) represents not instantiating the security VNF when the network is under attack.

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

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

Model performance validation was implemented using 5-fold cross-validation, i.e., five different measurements validate the implemented models, assessing their generalization ability of predictive and preventing overfitting. In addition, all data instances used in testing step were distinct from those used for training the models, as explained on Subsection V-A. We also defined two placement strategies for analyzing the ML models performance. They are presented on Subsections V-B and V-C.

A. Data Set Composition

In order to compose a realistic IoT network scenario that would match the performance indicators considered in Section III, we created a new data set[2] composed from two

[1]https://github.com/werneckg/master/tree/master/experiments/code

[2]https://github.com/werneckg/master/tree/master/experiments/dataset

TABLE IV: Qualitative rational to include performance indicators into data set.

Performance indicator	Network response time [16]	
	Low	High
CPU consumption (MHz)	High	Low
RAM consumption (MB)	High	Low
Environment deployment time (s)	Low	High
ML model accuracy (%)	Low	High
Approach response time (s)	Low	High

TABLE V: Performance indicators considered to train and test ML models.

Category	Performance indicator	Origin
NFV	CPU consumption (MHz)	Works in literature [33], [23], [30], [29], [26], [25], [32], [27], [31], [28]
	RAM consumption (MB)	
	Environment deployment time (s)	
Solution	Approach response time (s)	
	ML model accuracy (%)	
Network	Latency (s)	Fog IIoT Factory data set [16]
	Throughput (Mbps)	
	Network response time (s)	
	Number of processed packets	BoT-IoT data set [17]
	Size of processed packets (Bytes)	
	Application ports	
	Packet rates	

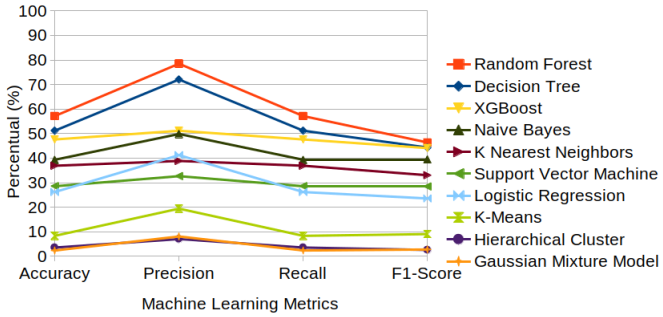
existing ones. Table V summarizes the indicators and their origins. The first refers to a Fog IIoT Factory [16] and the second to BoT-IoT [17]. Fog IIoT Factory Scenario is based on fog computing to support the QoS requirements of IIoT devices and describes details needed to predict applications' response time in this context. It consists of IIoT devices installed in robots and fog servers that provide computing services to devices. This data set provides Network performance indicators (i), such as latency, throughput and others.

BoT-IoT data set includes legitimate and simulated IoT network traffic, along with various types of attacks. We focus on DDoS attacks in this work, considering three types present in BoT-IoT: TCP, UDP and HTTP. Initially, (ii) we extract all traffic features incorporated in BoT-IoT testbed from random instances. So, we filter DDoS attacks and randomly pick the traffic generated by the testbed. Also, we selected benign traffic to evaluate the implemented models behavior in decision whether or not to instantiate a security VNF.

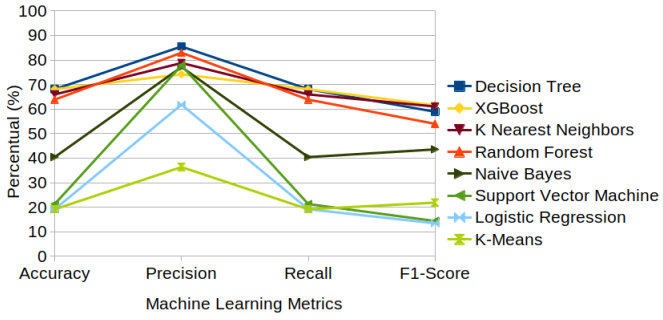
Closing the proposed network scope, (iii) we include performance indicators related to NFV and Solution disseminated in the literature, presented in Section III. Related to NFV, three related works refer to CPU and RAM consumption and two others to environment deployment time. Related to Solution, three works refer to ML models accuracy and six others to approach response time. Based on this analysis, we included these performance indicators in the new data set, taking as reference the predicted response time for IIoT network scenarios presented in [16]. Table IV resumes this qualitative rational. Each performance indicator was compared with the network response time individually. For example, the higher approach response time, the longer the network response time. They are proportionate. Conversely, the more computing power, CPU and RAM, the shorter the network response time.

B. Security VNF Deployed Near the Attacked IoT Device

As any IoT device is susceptible to DDoS attacks in the network scenarios, we analyzed how the ML models were able to classify the security VNF placing specifically on fog servers near the attacked IoT device, as illustrated in Fig. 1. Fig. 2 summarizes both supervised and unsupervised ML models performance. According to accuracy, all models had low performance, on average 30%. As for unsupervised models, the average was even lower, around 5%. This behavior is similar in both training and testing data. It is also extended to precision, recall and F1-score metrics.



(a) Training performance.



(b) Testing performance.

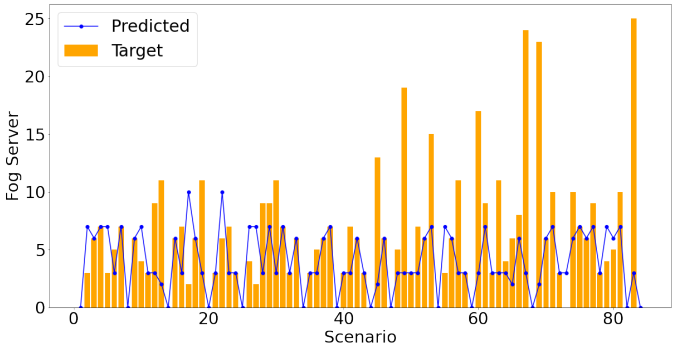
Fig. 2: Performance when allocating near the attacked device.

Focusing on supervised models, it is possible to observe, in Figs. 3a, 3b, 4a, 4b, 5a and 5b, the results obtained by the best models. Each point in the vertical axis represents a specific fog server. The orange bars represent the ideal fog server where to instantiate the security VNF. The blue points/lines represent the fog server selected by the ML model to instantiate the security VNF (A perfect model would have all the blue points on the top of all the orange bars). Each point of the horizontal axis represents a specific IIoT network scenario. Scenarios that have only normal traffic, without DDoS attacks, are the scenarios without a bar in the graph, i.e. the fog server identified by zero in fact does not exist. It just represents the case where no instantiation is needed. The models performed poorly mainly on network scenarios that predicted more fog servers, starting from network scenario 45.

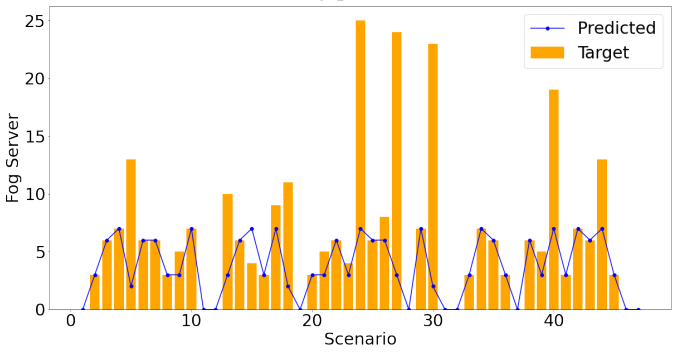
Going deeper and analyzing the distribution of fog servers in which the security VNFs are positioned to carry out models training, it is possible to observe a certain imbalance in data distribution. Most network scenarios predicted the security

VNF placement up to fog server 11. In fact, this imbalance contributes to the negative models performance. However, this is not the only factor that contributes to the low performance of unsupervised models. In this case, not labeling the target fog server negatively affects the training step. The classification is not adjusted according to the correct security VNF placement.

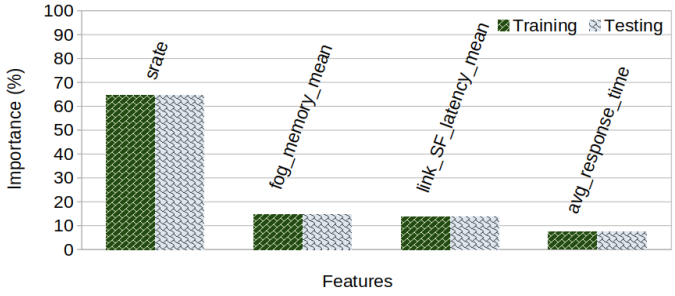
Another analysis performed was about feature importance. Figs. 3c, 4c and 5c summarize them for Decision Tree, Random Forest, and XGBoost, respectively. Overall, the features that most contributed to the models were Network-related, such as srate (source-to-destination packets per second), average link latency, standard deviation and average response time, bytes (total number of bytes in transaction) and NFV-related, such as CPU consumption.



(a) Training performance.



(b) Testing performance.

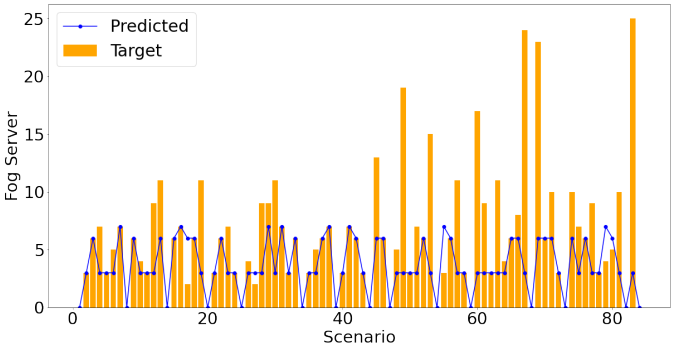


(c) Features importance for training and testing data.

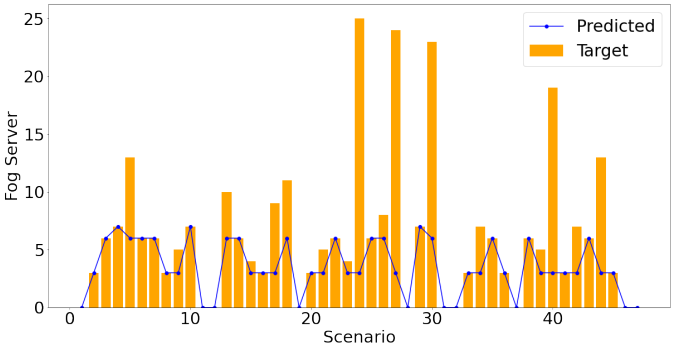
Fig. 3: Decision Tree classification performance and features importance when allocating near the attacked device.

C. Security VNF Deployed Near the Attacker Device

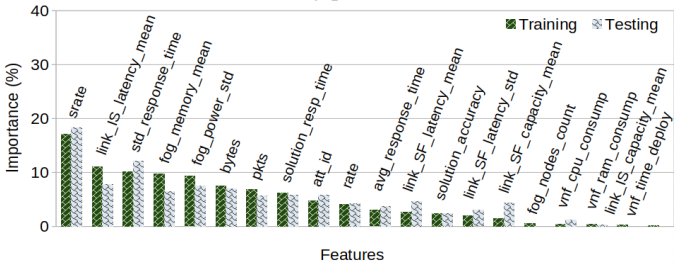
Another placement strategy is to instantiate the security VNF closer to attackers, also illustrated in Fig. 1. This subsection reports the results obtained when considering this strategy. Fig. 6 summarizes the ML models performance. When compared to the previous placing strategy, it is possible to notice a significant improvement in all metrics: accuracy, precision, recall and F1-score. This increase is about 19% overall for both training and testing data. The unsupervised models follow the same behavior as the previous experiments, that is, they have low performance in all metrics. On average, they had 24% of accuracy, a significant decrease in relation to supervised models that obtained an average of 71% of accuracy.



(a) Training performance.



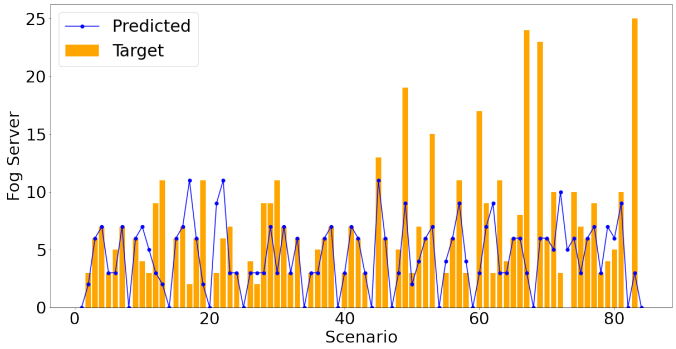
(b) Testing performance.



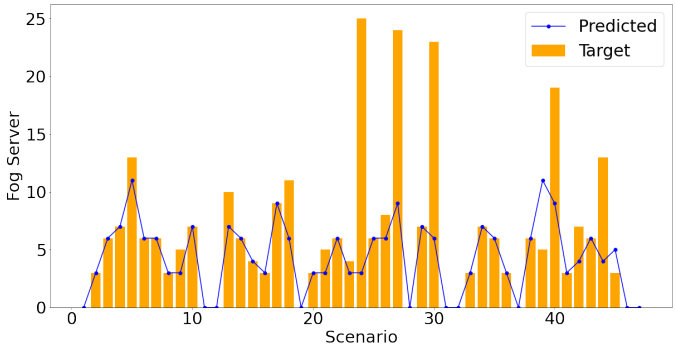
(c) Features importance for training and testing data.

Fig. 4: Random Forest classification performance and features importance when allocating near the attacked device.

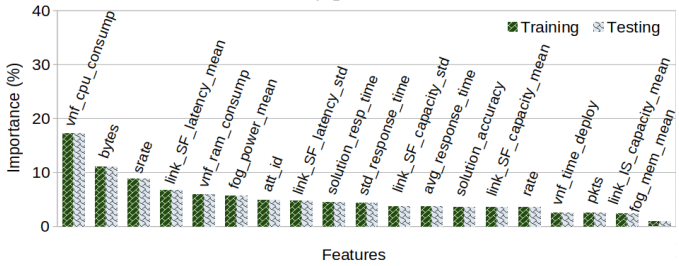
It is possible to highlight the supervised models XGBoost, Decision Tree and Random Forest. They had, respectively, average accuracies of 99.40%, 94.08% and 88.94% for training and testing steps. Figures 7c, 8c and 9c illustrates the features importance for them. The performance indicators that contributed the most to the models were related to Network and NFV. It is worth noting for XGBoost, the model with the best performance, the features that stood out the most were



(a) Training performance.



(b) Testing performance.

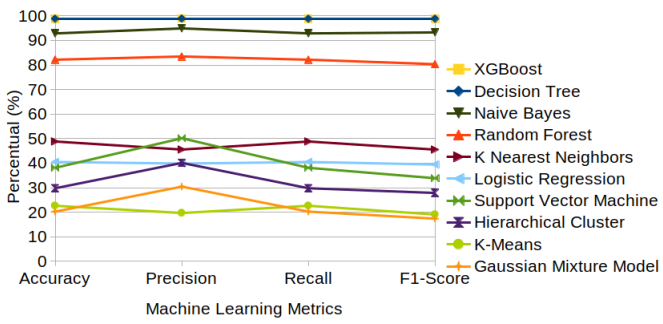


(c) Features importance for training and testing data.

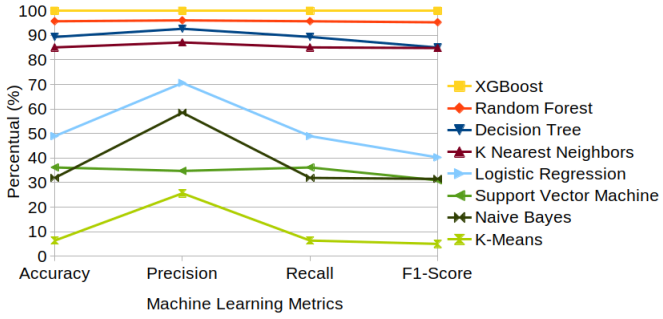
Fig. 5: XGBoost classification performance and features importance when allocating near the attacked device.

VNF CPU consumption and deployment time, with a total importance of about 22%. Also, the Network-related features that most influenced the model were srate, link capacity mean, standard deviation and average response time. This behavior is also seen in the previous placement strategy.

One factor that positively contributes to the performance achieved by the models is the reduced number of fog servers on which the security VNF can be deployed. In this case, as the attack instances were extracted from BoT-IoT, only four attackers could generate DDoS traffic. Consequently, only four fog servers were enabled to instantiate the security VNF. Figs. 7a, 7b, 8a, 8b, 9a and 9b, show how the three best models behave in detail. The models present a satisfactory performance. According to the plots and also to recall (3) metric, the models do not fail to instantiate the security VNF when DDoS attacks are present on the network. Errors made by the models are related to wrong instantiation when there is no attack on the network, that is, false positives. The issue in this case would be false negatives, as they would make the



(a) Training performance.



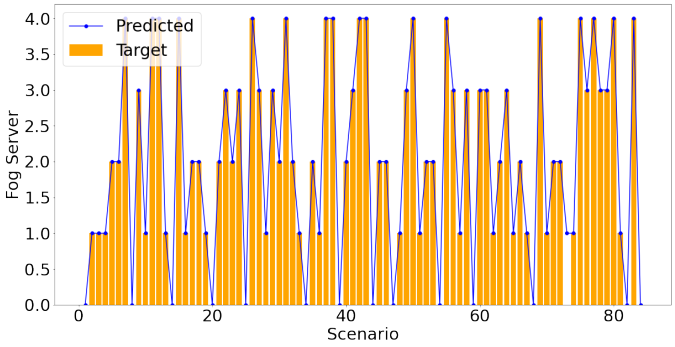
(b) Testing performance.

Fig. 6: Performance when allocating near the attacker device.

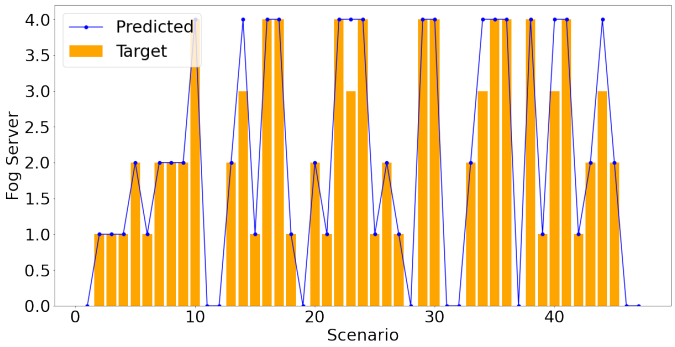
models to not instantiate the security VNF in scenarios under attacks, something that is not observed in the experiments.

VI. Conclusion

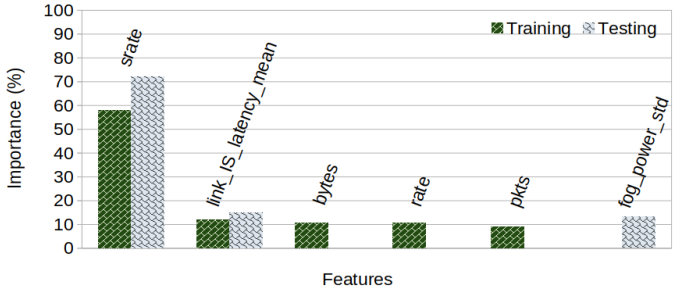
This article proposed and evaluated ML models used to place security VNFs on IIoT network, by considering performance indicators. The experiments carried out showed that models implementing tree-based structures, XGBoost, Decision Tree and Random Forest, are the most suitable for the proposed network scenario, achieving accuracy of 99.40%, 94.08% and 88.94%, respectively. Indeed, solution-related performance indicators are important when the objective of a security VNF is to mitigate attacks on IoT networks. However, they are not unique. It is necessary to consider other indicators related to Network and NFV to compose more efficient solutions. This work contributes to this, showing how context indicators of IIoT networks can be used to compose ML classification based solutions. The different sizes of network scenarios do not directly influence ML models training and testing time presented in this work, as what actually influences this time would be the number of scenarios for the models. What directly influences ML models accuracy is the number of fog servers. Furthermore, placing security VNFs close to the attacker demonstrates to be the best strategy for the IIoT network scenario presented here. Not only because of the models performance presented, but also because it does not allow a large flood of network packets to occur, unlike placing close to the IoT device. In other words, the further away the DDoS attack mitigation is from the attacker, the more malicious packets travel through the network until reach the target.



(a) Training performance.

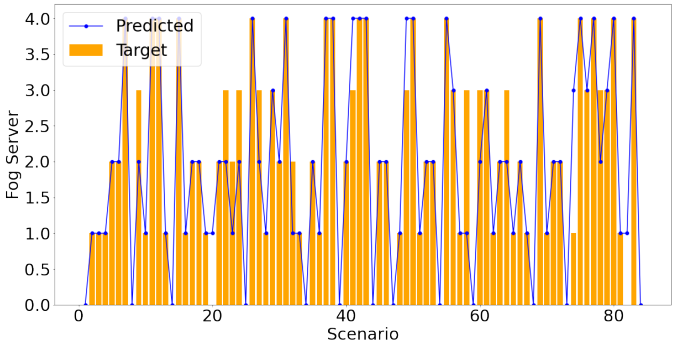


(b) Testing performance.

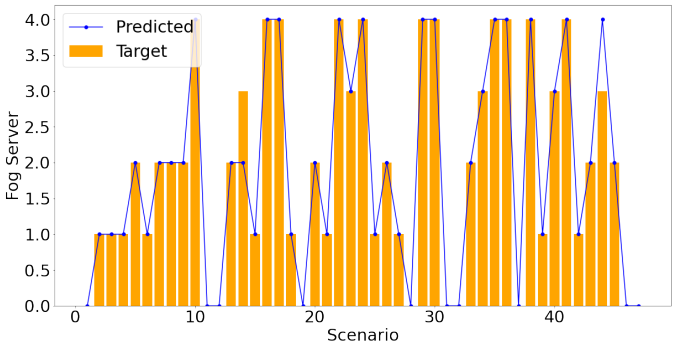


(c) Features importance for training and testing data.

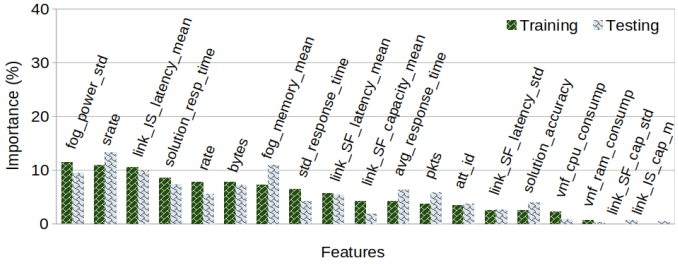
Fig. 7: Decision Tree classification performance and features importance when allocating near the attacker device.



(a) Training performance.



(b) Testing performance.



(c) Features importance for training and testing data.

Fig. 8: Random Forest classification performance and features importance when allocating near the attacker device.

ACKNOWLEDGMENT

This work was developed with support of the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). We thank the research agency for the financial proc. 130762/2021-0. The research is part of the INCT of the Future Internet for Smart Cities funded by CNPq proc. 465446/2014-0, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001, FAPESP proc. 14/50937-1 and proc. 15/24485-9. It is also part of the FAPESP proc. 18/23098-0.

REFERENCES

[1] M. de V. D. da Silva, A. Rocha, R. L. Gomes, and M. Nogueira, "Lightweight Data Compression for Low Energy Consumption in Industrial Internet of Things," in *Prof. of the IEEE CCNC*, 2021.

[2] D. M. Batista, A. Goldman, R. Hirata, F. Kon, F. M. Costa, and M. Endler, "InterSCity: Addressing Future Internet research challenges for Smart Cities," in *Proc. of the 7th NoF*, 2016, pp. 1–6.

[3] D. Rosário, Z. Zhao, A. Santos, T. Braun, and E. Cerqueira, "A Beaconless Opportunistic Routing based on a Cross-layer Approach for Efficient Video Dissemination in Mobile Multimedia IoT Applications," *Computer Communications*, vol. 45, pp. 21–31, 2014.

[4] J. Cui, F. Wang, Q. Zhang, C. Gu, and H. Zhong, "Efficient batch authentication scheme based on edge computing in iiot," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2022.

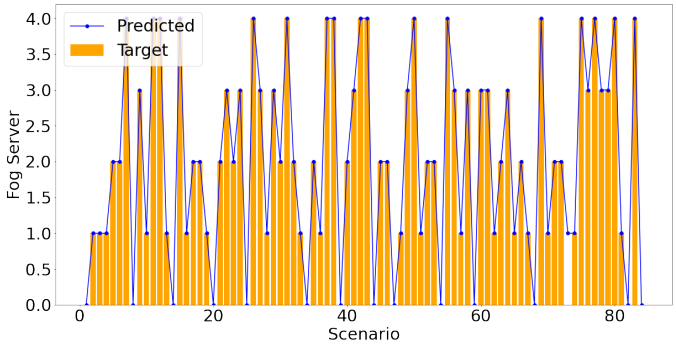
[5] I. Alam, K. Sharif, F. Li, Z. Latif, M. M. Karim, S. Biswas, B. Nour, and Y. Wang, "A Survey of Network Virtualization Techniques for Internet of Things Using SDN and NFV," *ACM Computing Surveys*, vol. 53, no. 2, pp. 1–40, 2020.

[6] Z. Chi, Y. Li, H. Sun, Y. Yao, and T. Zhu, "Concurrent Cross-Technology Communication among Heterogeneous IoT Devices," *IEEE/ACM TON*, vol. 27, no. 3, pp. 932–947, 2019.

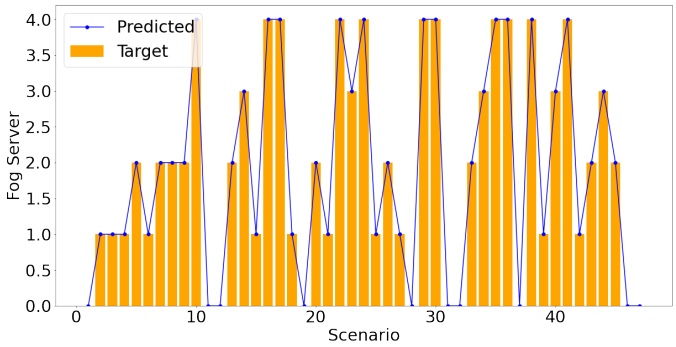
[7] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices," *IEEE IoT Journal*, vol. 6, no. 5, pp. 8182–8201, 2019.

[8] H. Griffioen and C. Doerr, "Examining mirai's battle over the internet of things," in *Proc. of the ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '20. New York, NY, USA: ACM, 2020, p. 743–756.

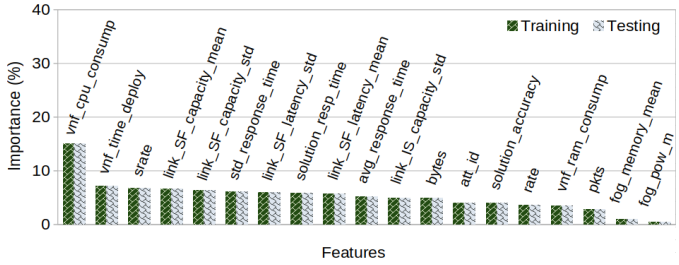
[9] Kaspersky, "Almost 30% of companies using IoT have experienced security incidents," 2020. [Online]. Available: https://www.kaspersky.com.br/blog/empresas-iot-seguranca-dicas/14869



(a) Training performance.



(b) Testing performance.



(c) Features importance for training and testing data.

Fig. 9: XGBoost classification performance and features importance when allocating near the attacker device.

[10] CheckPoint, "Keeping the gate locked on your IoT devices: Vulnerabilities found on Amazon's Alexa," 2020. [Online]. Available: https://research.checkpoint.com/2020/amazons-alexa-hacked

[11] I. Farris, T. Taleb, Y. Khettab, and J. Song, "A Survey on Emerging SDN and NFV Security Mechanisms for IoT Systems," *IEEE COMST*, vol. 21, no. 1, pp. 812–837, 2019.

[12] A. Laghrissi and T. Taleb, "A Survey on the Placement of Virtual Resources and Virtual Network Functions," *IEEE COMST*, vol. 21, no. 2, pp. 1409–1434, 2019.

[13] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. D. Turck, and R. Boutaba, "Network Function Virtualization: State-of-the-Art and Research Challenges," *IEEE COMST*, vol. 18, no. 1, pp. 236–262, 2016.

[14] L. Gupta, T. Salman, M. Zolanvari, A. Erbad, and R. Jain, "Fault and Performance Management in Multi-Cloud Virtual Network Services using AI: A Tutorial and a Case Study," *Computer Networks*, vol. 165, p. 106950, 2019.

[15] W. Zhang, J. Hwang, S. Rajagopalan, K. Ramakrishnan, and T. Wood, "Performance Management Challenges for Virtual Network Functions," in *2016 IEEE NetSoft Conference and Workshops*, 2016, pp. 20–23.

[16] G. W. de Oliveira, R. Ney, J. Herrera, D. M. Batista, R. H. Jr., J. Galán-Jiménez, J. Berrocal, J. M. Murillo, A. L. dos Santos, and M. Nogueira, "Predicting Response Time in SDN-Fog Environments for IIoT Applications," in *Proc. of the IEEE LATINCOM*, 2021.

[17] N. Moustafa, "The bot-iot dataset," 2019. [Online]. Available: https://dx.doi.org/10.21227/r7v2-x988

[18] ETSI, "Network Functions Virtualisation (NFV): NFV Performance & Portability Best Practises," 2014. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV-PER/001_099/001/01.01.01_60/gs_NFV-PER001v010101p.pdf

[19] ——, "Network Functions Virtualisation (NFV): Service Quality Metrics," 2014. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV-INF/001_099/010/01.01.01_60/gs_NFV-INF010v010101p.pdf

[20] IETF, "Benchmarking Methodology for Virtualization Network Performance," 2017. [Online]. Available: https://tools.ietf.org/id/draft-huang-bmwg-virtual-network-performance-03.html

[21] T. Kim, T. Koo, and E. Paik, "SDN and NFV Benchmarking for Performance and Reliability," in *Proc. of the APNOMS*, 2015, pp. 600–603.

[22] IETF, "RFC 8172: Considerations for Benchmarking Virtual Network Functions and Their Infrastructure," 2017. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc8172

[23] A. M. Zarca, J. B. Bernabe, I. Farris, Y. Khettab, T. Taleb, and A. Skarmeta, "Enhancing IoT security through network softwarization and virtual security appliances," *International Journal of Network Management*, vol. 28, no. 5, p. e2038, 2018, e2038 nem.2038.

[24] I. Farris, J. B. Bernabe, N. Toumi, D. Garcia-Carrillo, T. Taleb, A. Skarmeta, and B. Sahlin, "Towards provisioning of SDN/NFV-based security enablers for integrated protection of IoT systems," in *Proc. of the IEEE CSCN*, 2017, pp. 169–174.

[25] A. M. Zarca, D. Garcia-Carrillo, J. B. Bernabe, J. Ortiz, R. Marin-Perez, and A. Skarmeta, "Enabling Virtual AAA Management in SDN-Based IoT Networks," *Sensors*, vol. 19, no. 2, 2019.

[26] A. M. Zarca, J. B. Bernabe, R. Trapero, D. Rivera, J. Villalobos, A. Skarmeta, S. Bianchi, A. Zafeiropoulos, and P. Gouvas, "Security Management Architecture for NFV/SDN-Aware IoT Systems," *IEEE IoT Journal*, vol. 6, no. 5, pp. 8005–8020, 2019.

[27] M. Bagaa, T. Taleb, J. B. Bernabe, and A. Skarmeta, "A Machine Learning Security Framework for Iot Systems," *IEEE Access*, vol. 8, pp. 114 066–114 077, 2020.

[28] A. M. Zarca, J. B. Bernabe, A. Skarmeta, and J. M. A. Calero, "Virtual IoT HoneyNets to Mitigate Cyberattacks in SDN/NFV-Enabled IoT Networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1262–1277, 2020.

[29] R. Sairam, S. S. Bhunia, V. Thangavelu, and M. Gurusamy, "NETRA: Enhancing IoT Security Using NFV-Based Edge Traffic Analysis," *IEEE Sensors Journal*, vol. 19, no. 12, pp. 4660–4671, 2019.

[30] A. Boudi, I. Farris, M. Bagaa, and T. Taleb, "Assessing Lightweight Virtualization for Security-as-a-Service at the Network Edge," *IEICE Transactions on Communications*, vol. E102.B, no. 5, pp. 970–977, 2019.

[31] N. Guizani and A. Ghafoor, "A Network Function Virtualization System for Detecting Malware in Large IoT Based Networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1218–1228, 2020.

[32] L. Zhou, H. Guo, and G. Deng, "A fog computing based approach to DDoS mitigation in IIoT systems," *Computers & Security*, vol. 85, pp. 51–62, 2019.

[33] M. Al-Shaboti, I. Welch, A. Chen, and M. A. Mahmood, "Towards Secure Smart Home IoT: Manufacturer and User Network Access Control Framework," in *Proc. of the IEEE 32nd AINA*, 2018, pp. 892–899.

[34] S. K. Moghaddam, R. Buyya, and K. Ramamohanarao, "Performance-Aware Management of Cloud Resources: A Taxonomy and Future Directions," *ACM Computing Surveys*, vol. 52, no. 4, Aug. 2019.

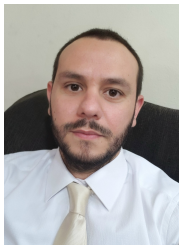
[35] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley, 1991.

[36] S. Bhulai, S. Sivasubramanian, R. van der Mei, and M. van Steen, "Modeling and Predicting End-to-End Response Times in Multi-tier Internet Applications," in *Managing Traffic Performance in Converged Networks*, L. Mason, T. Drwiega, and J. Yan, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 519–532.

[37] S. Yang, F. Li, S. Trajanovski, X. Chen, Y. Wang, and X. Fu, "Delay-aware virtual network function placement and routing in edge clouds," *IEEE Transactions on Mobile Computing*, vol. 20, no. 2, pp. 445–459, 2021.

[38] S. Song, C. Lee, H. Cho, G. Lim, and J.-M. Chung, "Clustered virtualized network functions resource allocation based on context-aware grouping in 5g edge networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 5, pp. 1072–1083, 2020.

[39] Y.-Y. Shih, H.-P. Lin, A.-C. Pang, C.-C. Chuang, and C.-T. Chou, "An nfv-based service framework for iot applications in edge computing environments," *IEEE TNSM*, vol. 16, no. 4, pp. 1419–1434, 2019.



Guilherme Werneck de Oliveira is Bachelor of Computer Science from Federal University of Itajubá (UNIFEI) and M.Sc. in Computer Science from the University of São Paulo (USP). His main research interests include network functions virtualization, performance management, security, machine learning and Internet of Things networks.

Michele Nogueira is an associate professor in the Computer Science Department of Federal University of Minas Gerais (UFMG). She holds a Doctorate degree in Computer Science from the UPMC-Sorbonne University, Laboratoire d'Informatique de Paris VI (LIP6). She was in a Sabbatical Leave at Carnegie Mellon University in 2016-2017. Her research interests include wireless networks, network security and resilience. She was an Associate Technical Editor for the IEEE Communications Magazine and the Journal of Network and Systems Management.

Aldri L. dos Santos is professor of the Department of Computer Science at Federal University of Minas Gerais (UFMG). Aldri is PhD in Computer Science from the Federal University of Minas Gerais, Master in Informatics and Bachelor of Computer Science at UFPR. Aldri has worked in the following research areas: network management, fault tolerance, security, data dissemination, wireless ad hoc networks and sensor networks. He is leader of the research group (Wireless and Advanced Networks). Aldri has also acted as reviewer for publications as IEEE ComMag, IEEE ComNet, ComCom, IEEE Communications Surveys and Tutorials, IEEE eTNSM, JNSM, Ad hoc Networks. Aldri has served as member of the technical committee of security information and IEEE Communication Society Communication (ComSoc).

Daniel Macêdo Batista is an Associate Professor at the University of São Paulo (USP) in the Department of Computer Science. He is PhD and Master in Computer Science from the State University of Campinas (Unicamp). He received his Computer Science Bachelor's degree from Federal University of Bahia (UFBA). He already served as member of the editorial board of IEEE Communications Surveys and Tutorials and is a regular reviewer for IEEE TNSM, IEEE Network, IEEE TSC, Computer Networks and Journal of Supercomputing. Since 2022 he is the secretary of the Internet Technical Committee. Currently, his main research interests are: IoT Security, B5G, and Data Analytics applied to Computer Networks.