Received March 31, 2020, accepted April 21, 2020, date of publication May 4, 2020, date of current version May 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2992044

Long Short-Term Memory and Fuzzy Logic for Anomaly Detection and Mitigation in Software-Defined Network Environment

MATHEUS P. NOVAES[1], LUIZ F. CARVALHO[2], JAIME LLORET[3], (Senior Member, IEEE), AND MARIO LEMES PROENÇA, Jr.[4]

[1]Electric Engineering Department, State University of Londrina (UEL), Londrina 86057-970, Brazil
[2]Computer Engineering Department, Federal Technology University of Paraná (UTFPR), Apucarana 86812-460, Brazil
[3]Integrated Management Coastal Research Institute, Universitat Politecnica de Valencia, 46022 Valencia, Spain
[4]Computer Science Department, State University of Londrina (UEL), Londrina 86057-970, Brazil

Corresponding author: Mario Lemes Proença, Jr. (proenca@uel.br)

This work was supported in part by the National Council for Scientific and Technological Development (CNPq) of Brazil under Project 310668/2019-0, in part by the SETI/Fundação Araucária due to the concession of scholarships, and in part by the Ministerio de Economía y Competitividad through the Programa Estatal de Fomento de la Investigación Científica y Técnica de Excelencia, Subprograma Estatal de Generación de Conocimiento, under Grant TIN2017-84802-C2-1-P.

ABSTRACT Computer networks become complex and dynamic structures. As a result of this fact, the configuration and the managing of this whole structure is a challenging activity. Software-Defined Networks(SDN) is a new network paradigm that, through an abstraction of network plans, seeks to separate the control plane and data plane, and tends as an objective to overcome the limitations in terms of network infrastructure configuration. As in the traditional network environment, the SDN environment is also liable to security vulnerabilities. This work presents a system of detection and mitigation of Distributed Denial of Service (DDoS) attacks and Portscan attacks in SDN environments (LSTM-FUZZY). The LSTM-FUZZY system presented in this work has three distinct phases: characterization, anomaly detection, and mitigation. The system was tested in two scenarios. In the first scenario, we applied IP flows collected from the SDN Floodlight controllers through emulation on Mininet. On the other hand, in the second scenario, the CICDDoS 2019 dataset was applied. The results gained show that the efficiency of the system to assist in network management, detect and mitigate the occurrence of the attacks.

INDEX TERMS DDoS, deep learning, fuzzy, LSTM, portscan, SDN.

I. INTRODUCTION

Nowadays, the number of applications and services that use the Internet has increased quickly. The network system has become complex structures due to a large number of devices that make them, for example, firewall, intrusion detection system, load balancer, switches, routers, etc. In the traditional network environment, each network asset uses complex protocols, and its configuration differs between makers. With the advent of Cloud Computing and the increase in virtualization technologies, the traditional management architecture of the network is not adequate for these applications, particularly at the current data centers [1].

Despite Software-Defined Networking (SDN) not having been created with a specific objective for virtualization functions of the network, it is an emerging network architecture that projects future networks and that meets the new demands of already existing applications [2], [3]. The main characteristic of SDN architecture is the separation of the control and data plane, which means that the control plane is removed from the network device and centralized on a controller [4], [5]. The centralization of the control plane provides a global view of the network and allows the management of its components through an open and well-defined software interface [6].

Along with the increased demand for web applications and the popularization of new IoT (Internet of Things) devices, issues related to security emerge, for example, attacks [7], [8]. The number of attacks has increased in numbers and in the sophistication of how they are carried out by malicious agents, especially the Distributed Denial of Services (DDoS). The purpose of DDoS is to exhaust a resource, even at the

The associate editor coordinating the review of this manuscript and approving it for publication was Amjad Ali.

server level where the attacker, through many solicitations, tries to deprive some service or at the infrastructure level where the attacker saturates a network link [9], [10].

Although the SDN networks have introduced programming and centralization resources of the control logic, which facilitate its management, these resources are the main security vulnerabilities presented by the network architecture [11], [12]. Due to the SDN network architecture, it is known that the management of network flows is centralized and executed by a controller, which is subject to security threats, for example, DDoS attacks, Portscan, IP spoofing, etc [10], [13]–[16]. During a DDoS attack, the network services are overloaded due to the large number of requests sent by the attackers. The controller is the central point of the SDN network and is vulnerable to attacks. Besides, DDoS attacks are followed by Portscan attacks, where the attacker scans for open ports to perform intrusions. Thus, the SDN network security remains undefined and it is necessary the development of solutions related to detection and mitigation of attacks.

In the last few years, with the increase of security threats and the huge enormous volume of traffic, several approaches have been proposed to detect anomalies [17]–[19]. Network anomaly detection consists of two main approaches: Signature-based and Profile-based. For the first one, Signature-based, a database containing signatures of the most diverse kinds of attacks is needed, and the detection of an anomalous event occurs when there is a ''math'' between the behavior of the network and the known pattern attack. On the other hand, the profile-based approach, based on network history data, a prediction of its usual behavior is made, and an anomaly is detected when the predicted behavior and the real behavior diverge from one another [20]. One of the main advantages of this kind of methodology is the detection of unknown anomalies, for the system does not require learning the behavior of the many existing attacks. Furthermore, the current attacks are dynamic, and new patterns emerge frequently [7], [17].Thus, Signature-based approaches demand that the signature of the attacks are updated each time a new attack emerges, resulting in a drawback for the system.

Generally, the anomaly detection techniques intend to recognize sensitive traffic patterns through sudden changes in the expected traffic volume or unexpected changes in the distribution of specific network traffic characteristics, such as IP addresses and ports. The implementation of Machine Learning algorithms provides solutions for detecting and classifying anomalies [21], [22]. These algorithms have the capacity of learning patterns from a set of data and making predictions based on these data. Usually, the Machine Learning techniques employed in anomaly detection systems are divided into two approaches: Shallow Learning and Deep Learning. Shallow Learning algorithms have some limitations, such as largely depending on attributes used in the process of training, and an intensive analysis is necessary in order to capture the most relevant attributes and statistics of the traffic [23], [24]. Besides, the models often need to be retrained to learn new patterns of network behavior [25], [26]. Recently, the methods based on Deep Learning have been applied in many works related to intrusion detection systems, due to the learning capacity and generalization of employed attributes [27]–[29].

Thus, we present a modular system for anomaly detection and mitigation applied on SDN networks environments. The developed system consists of three modules with well-defined functions. The first module is the characterization one, which employs a Deep Learning algorithm called Long short-term memory (LSTM), an architecture of artificial recurrent neural network (RNN), to predict the normal behavior of the network traffic. The second module is responsible for detecting anomalous events, in which the Bienaymé-Chebyshev inequality is applied to generate normality threshold dynamically, and with that, the Fuzzy logic is applied to identify the occurrence of an anomaly in a certain moment of the analysis. The third module of the system is responsible for the mitigation of detected anomalies, intending to minimalize the damages caused by an attacker.

The main contributions of this work are:

- Network traffic characterization employing a Deep Learning LSTM mechanism;
- DDoS and Portscan attacks detection using a Fuzzy inference system;
- Analysis of the network traffic performed in near real-time;
- Test with two datasets containing many kinds of DDoS attacks;
- Comparison between the developed system and other methods present in the literature.

The rest of this work is organized as the following: Section II presents the related works; Section III presents the fundamentals used in the development of the system; Section IV we discuss the system performance results. Ultimately, on Section V the conclusions obtained with the development of this paper is presented.

II. RELATED WORKS

Nowadays, SDN networks are used broadly, however they present many problems related to security [4], [6], [13], [30]. Thus, the SDN network security remains indefinite, and solutions related to detection and mitigation of attacks have been developed [14].

According to AlEroud and Alsmadi [31], when the packet forwarding logic is centralized and allocated in the controller, the malicious agents explore vulnerabilities on the controller, on links of communication between controller and forwarding devices and on switches' memory. A switch has a limited memory, when it is under attack, the number of flows received by the devices increase considerably, taking up all the storage capacity from the forwarding table. Many studies have been developed in order to create defense mechanisms to supply these vulnerabilities present in SDN architecture [7], [32]. Silva *et al.* [33] introduced a framework called ATLANTIC

(*Anomaly deTection and machine LeArNing Traffic classifIcation for software-defined networking*) for detecting, classifying and mitigating anomalous events in SDN networks. Garg and Garg [34] present an adaptive mechanism to update the policies dynamically to aggregate the flows inputs and to detect anomalies, so that the monitoring overcharge can be reduced and the anomalies can be detected more precisely. Mousavi and St-Hilaire [35] applied a technique to DDoS detection using entropy. The main goal of the authors is to detect an attack on its first stage, for the detection made at the beginning of the attack allows the application of mitigation policies before the controller is completely flooded with malicious packets.

Carvalho *et al.* [36] presents a new ecosystem to detect and mitigate DDoS attacks in SDN environments. The system proposed by the authors is composed by four stages: the first one is related to collection and storing of IP flow records; the second stage is the generation of a normal network profile based on the data collected using the ACODS method (*Ants Colony Optimization for Digital Signature*); the third stage corresponds to the detection of anomalies comparing the real network behavior to the generated profile using multinomial logistic regression (MLR) to detect suspicious events that differ from the expected behavior; finally, in the fourth stage mitigation policies are applied. The analysis of the traffic behavior for anomaly detection is done every 30 seconds. According to the results presented by the authors, the system proves to be efficient at the detection and mitigation of anomalous events stages.

Hamamoto *et al.* [37] proposed a system of anomalies detection applied to large scale networks. The authors used the DSNSF approach (*Digital Signature of Network Segment using Flow Analysis*) to generate behavior signatures of normal network behavior, applying GA (Genetic Algorithm). Furthermore, the Fuzzy logic was used along with DSNSFs generated to anomalous behaviors in those analyzed networks. It was used real data collected from the State University of Londrina by using sFlow to validate the proposed system. Three different anomalies were injected into the network's real data, using tools for simulation of anomalous events: DoS, DDoS, and Flash Crowd. The suggested system showed to be efficient, with a prediction rate above 96%. Different works also applied the DSNSF approach by using different techniques. However, the traffic characterizations on these works used an approach which analyses from two to four weeks of data for recognizing patterns and generation of normal profile in the network's regular environment. Moreover, a limitation presented by these works is that the attacks were detected in the period between 1 and 5 minutes. Unlike these works, the model proposed in this paper performs the prediction of normal traffic behavior by applying a sliding window and detecting anomalous events every second.

With the increase of the applications of image recognition, natural language processing, bioinformatics, the Deep Learning models had a fundamental role in solving these kinds of problems. Due to its huge capacity to extract knowledge in large scale from complex data, obtaining advantages on its results if we compared them to the traditional Machine Learning techniques [18], [21]. In Cybersecurity, the Deep Learning models are being applied in many different areas, for example intrusion detection [38], malware detection, spam detection [39], DDoS attacks detection [40], etc.

Li *et al.* [41] proposed a supervised Learning Machine mechanism of defense and detection of DDoS attacks in SDN network environments based on deep learning. The model presented by the authors consists of the following layers: input layer, forward recursive layer, reverse recursive layer, fully connected hidden layer, and output layer. At the construction of the model, it was employed Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN). According to the result gained by this model after detection, the SDN controller generates discard policies and sends them to the switches. For the test conduction, the ISCX dataset was employed to train the detection and verification model of defense architecture through DDoS attacks in real-time. According to the presented results, the defense method presented obtained an accuracy rate of 98%. However, the supervised learning to detect network attacks is a drawback, because the way the attackers executed the attacks is constantly being updated.

Tuan A Tang *et al.* [42] employed Deep Learning to detect anomalous flows in the SDN network. The authors suggested a Deep Neural Network (DNN) for a system that detects intruders, and the model was trained by using the NSL-KDD dataset. The dataset is made of 41 attributes. However, only a subset of 6 attributes were used. Through experiments, the suggested model only obtained an accuracy of 75.75%. The low amount of attributes influenced the low accuracy. Dey and Rahman [43] present a method of anomalies detection based on flows on the OpenFlow controller using DNN. The suggested model combined two approaches Gated Recurrent Unit and Long Short Term Memory (GRU-LSTM) to construct the intrusion detection system. Two methods of feature selection were employed for each anomaly analyzed to improve the model's performance, the NOVA F-Test and Recursive Feature Elimination. For the experiment process, the NSL-KDD dataset was also used. The experimental results demonstrated an accuracy of 87%. Shone *et al.* [38] suggested a new DL ensemble model for NIDS, which combines deep and shallow learning. The model combines Non-symmetric Deep Auto-Encoder and Random Forest. The data used for the test came from KDD CUP 99 and NSL-KDD datasets. The results showed an accuracy of 97.85%.

Despite this, many works available in the literature [17], [21], [25], [26], [28] for detecting DDoS attacks only evaluate a few types of DDoS attacks. Unlike these works, one of the main contributions presented by the system proposed in this paper is the detection of 12 types of DDoS attacks (e.g., NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebD-DoS (ARME), SYN, and TFTP). In addition, the system proposed is capable of learning the normal

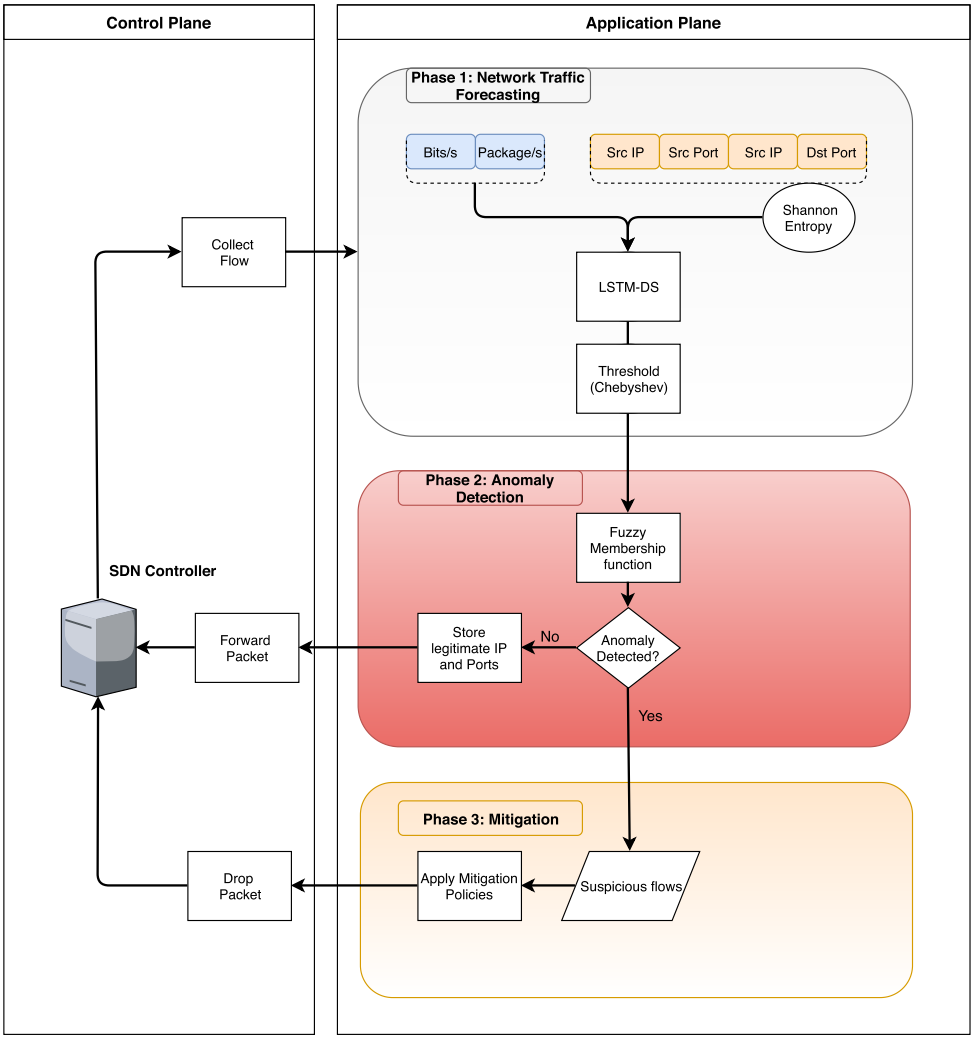

FIGURE 1. The proposed system architecture using LSTM and Fuzzy logic for Anomaly Detection and Mitigation.

behavior of the network, which is an advantage for detecting zero-day attacks.

III. THE SYSTEM PROPOSED

The system proposed in this paper has as its main goal the network traffic characterization, detection, and mitigation of DDoS attacks and Portscan in Software-defined networking environment. The system used as a principal the concept of Digital Signature of Network Segments (DSNS) introduced by Proença [20]. This concept applies an efficient technique to create a model that characterizes the network profile using historical data. The characterization proposed by Proença *et al*. was idealized for the traditional network environment and used a historical base from past traffic weeks containing MIB (Management Information Base) objects from management protocol SNMP (Simple Network Management Protocol). On the other hand, the characterization suggested in this paper uses IP flows attributes collected from the SDN controller, and the prediction of the network signature is made by employing a sliding window of the traffic. Consequently, the suggested system discards the use of a database to generate a signature. It is possible to recognize behavior from the normal profile of network that differs from the expected and helps in the anomaly detection stage methodology presented in this work.

The system of detection and mitigation of anomalies suggested in this work is divided in three phases:

1) Prediction of the normal behavior of the network's traffic and the definition of normality thresholds;
2) Application of the Fuzzy logic to determine if there are anomalies, using as a parameter the predicted traffic and the defined thresholds on the last stage;
3) Application of mitigation policies with the intention of taking countermeasures against the detected attacks, guarantying the network operation.

Fig. 1 illustrates the schematic diagram for operation of the anomaly detection and mitigation system suggested in this paper. The system was developed on the application

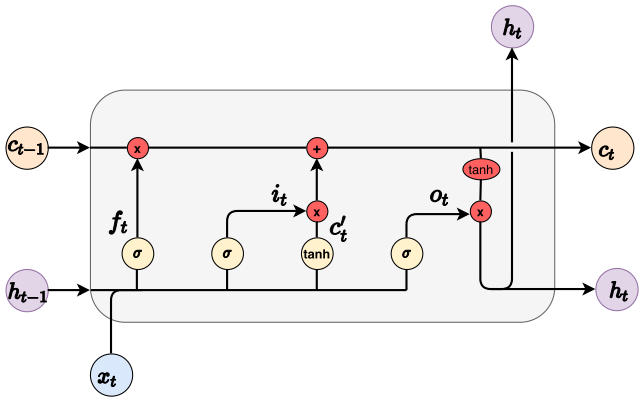

FIGURE 2. LSTM cell structure.

plane. The first stage is the traffic characterization, in which the flows attributes are pre-processed to predict the network traffic and the normal behavior signatures along with the normality thresholds. The next stage is the module of anomaly detection, in which the Fuzzy Inference system takes decision dynamically to determine the occurrence of anomalous events. When there is an anomaly, the IPs and ports that are in the analysis interval are considered suspicious. The third stage is responsible for the application of mitigation policies and receives as input suspicious flows determined on the last stage. In this set of flows, the mitigation module applies the most appropriate countermeasure to minimize the effects of an attack.

A. LONG SHORT-TERM MEMORY FOR NETWORK TRAFFIC FORECASTING

1) LONG SHORT-TERM MEMORY

In this subsection, some concepts about LSTM will be briefly introduced to assist in the understanding of the characterization module proposed in this work. Introduced by Hochreiter and Schmidhuber [44], LSTM a special architecture of recurrent artificial neural networks, with the capacity to learn long-term dependencies.

The structure of an LSTM cell is illustrated in Fig. 2. As observed, at each t time instant, the cell is controlled by various gates that can either maintain or reset the value according to the state of the gate. Three gates are applied on the cell, the forget gate (f_t), the input gate (i_t), and the output gate (o_t). Moreover, there is an entrance modulation gate called candidate value. The gates can be described as the following:

$$i_t = \sigma(W_{x,i}x_t + W_{i,h}h_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_{f,i}x_t + W_{f,h}h_{t-1} + b_f) \quad (2)$$

$$o_t = \sigma(W_{o,i}x_t + W_{o,h}h_{t-1} + b_o) \quad (3)$$

$$c'_t = tanh(W_{c',i}x_t + W_{c',h}h_{t-1} + b_{c'}) \quad (4)$$

where W means the matrix of synaptic weight, b means the bias vectors, x_t is the actual input, c'_t is a vector with new candidates to be added to the actual state of the cell, h_{t-1} is the LSTM previous output in the time of instant $t-1$, $\sigma(\cdot)$ and $\tanh(\cdot)$ are the respective activation functions, Sigmoid and Tangent Hyperbolic. The first step on LSTM is to decide how much of the previous memory value will be removed from the state of the cell. This decision is made by the forget gate. The next stage is to determine how much of the new information will be stored, which is made by the input gate. Next, the state of the cell is used and defined with the following expression:

$$c_t = f_t \odot c_{t-1} + i_t \odot c'_t \quad (5)$$

in which $\odot$ denotes elementwise product. The LSTM output h_t is defined by:

$$h_t = o_t \odot tanh(c_t) \quad (6)$$

2) NETWORK TRAFFIC FORECASTING PHASE

The traffic prediction aims to generate the network's normal behavior signature, which is essential for the management and for the network security. The network characterization makes the decision of management related to possible problems that may occur more reliable and safer. To obtain a prediction close to the real behavior is an important step towards the detection of anomalous traffic, for the generated signature delimits the normality limits of a traffic sample at a certain moment on the network segment that is observed.

The characterizations of the signatures are generated from IP Flow data that are collected from the SDN network switches by the controller using an OpenFlow protocol. Among the attributes collected by the controller, the following attributes were selected: bits/s, packets/s, source IP address, destination IP address, source and destination ports. These flows attributes were analyzed and employed to previous works in the network traffic characterization of high speed and presented good results to describe and better understand the network behavior [45], [46]. Bytes and packets dimensions are quantitative attributes, which means volume attributes that are capable of supplying information related to the amount of information that is transported on the network. The others are nominal attributes and supply qualitative information that means these attributes allow to understand which devices are communicating with one another and which services are being accessed by them. The use of these attributes is fundamental to identifying possible attacks and is indispensable for the use of module mitigation to minimize the damage caused by an attack.

The IP and port attributes are nominal data and to carry out a quantitative analysis it is necessary to transform these attributes through entropy calculus. So, the Shannon Entropy was used in this work [47], it allows information from the concentration to be extracted and the prediction of these flow attributes. Ultimately, with the set of flow attributes $x = \{x_1, x_2, \ldots, x_n\}$ where x_i represents the sample's occurrence i at the interval of time. The entropy H to X is defined in

the Equation (7)

$$H(x) = -\sum_{i=1}^{N} \left(\frac{x_i}{S}\right) log_2 \left(\frac{x_i}{S}\right), \quad (7)$$

in which $S = \sum_{i=1}^{N} x_i$ is the sum of all occurrences present in the histogram.

It is possible to identify attacks by using entropy to characterize traffic. For example, during a DDoS attack occurrence, there is a concentration of the victim's IP address and destination port entropy; dispersion of the entropy of the source port due to multiple attackers using random source ports.

After guarantying that all flow attributes collected are presented in a quantitative way, a process of traffic signature generation starts. The problem with the network traffic prediction using LSTM could be defined as the following model. Consider at the instant of time t, the set of data $\mathbf{X} = (x_1, x_2, \ldots, x_d)$, where each x_i is a flow attribute vector defined as:

- x_1: bits/s
- x_2: packets/s
- x_3: source IP entropy
- x_4: destination IP entropy
- x_5: source Port entropy
- x_6: destination Port Entropy

Long Short-Term Memory neural networks are designed to handle with sequence due to their ability to maintain long term memory. In recent years, LSTM is widely used in time series prediction and has proven to be superior to traditional mathematical algorithms [48]–[50]. Besides, LSTM is a powerful technique that can represent the relationship between current and previous events and enhance network traffic forecasting.

In the approach to this work, LSTM was applied to mold the problem of univariate time series prediction. In this way, LSTM predicts the signature of normal network behavior. An LSTM was applied to each flow attribute defined previously, which means each LSTM will be responsible for predicting the signature of normal behavior for each attribute x_i. The LSTM model will learn a function that maps a sequence of n observations of previous inputs to an output observation [51]. For example, at the t instant, given an input sequence of n past observations that consists of the bits vector $x_1 = \{x_{t-n}, \ldots, x_{t-3}, x_{t-2}, x_{t-1}\}$, produces an output $y_1 = \{y_t\}$ which represents the behavior prediction to the flow bits attribute. Fig. 3 illustrates the LSTM model For Digital Signature (LSTM-DS) proposed in this paper.

Despite using 6 LSTM networks, one for each flow attribute, the training process of the network is an offline task. The computing cost for the training is high. However, it is not critical to its application [52]. Being so, during the operation stage with the trained LSTM networks, the prediction process of traffic is immediate. The Algorithm 1 illustrates the process of the LSTM-DS module operation.

The predicted traffic would not be the same as the real traffic. However, it is necessary to determine

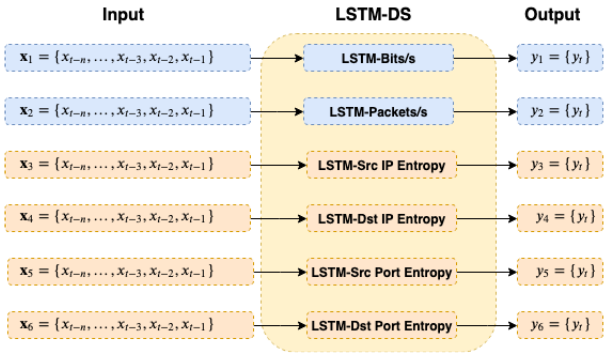

FIGURE 3. The proposed model for traffic forecasting using 6 LSTM.

Algorithm 1 LSTM-DS Operation Phase

Require: $\mathbf{X} = (x_1, x_2, \ldots, x_d)$
Ensure: $\mathbf{y} = (y_1, y_2, \ldots, y_d)$
1: $y_1 = \text{LSTM-bits}(x_1)$
2: $y_2 = \text{LSTM-Packets}(x_2)$
3: $y_3 = \text{LSTM-SrcIPEntropy}(x_3)$
4: $y_4 = \text{LSTM-DstIPEntropy}(x_4)$
5: $y_5 = \text{LSTM-SrcPortEntropy}(x_5)$
6: $y_6 = \text{LSTM-DstPortEntropy}(x_6)$
7: $\mathbf{y} = (y_1, y_2, y_3, y_4, y_5, y_6)$
8: **return y**
=0

thresholds between the predicted traffic and the real traffic. Bienaymé-Chebyshev's inequality is used to define this threshold between the predicted and the real one. Bienaymé-Chebyshev's inequality determines a limit of data percentage that lies in number k of standard deviations interval around the mean. The inequality can be applied to detect outliers [53] when the data distribution is unknown.

The equation for the Bienaymé-Chebyshev's inequality is represented in the Equation (8):

$$P(|\ X - \mu\ | \geq k\sigma) \leq \frac{1}{k^2}, \quad (8)$$

where X is a random variable, μ is the mean, $k > 0$ is the parameter of deviation and σ is the standard deviation. Defining the parameter $k = 4.47$ in Equation (8), the resulting probability will be equal to 0.05, which is the usual cut-off point for statistical significance to verify the discrepancy of a hypothesis in relation to the observed data [54].

B. FUZZY LOGIC FOR ANOMALY DETECTION

1) FUZZY LOGIC THEORY

In Classical logic, a proposition can only take values as true or false. On the other hand, the Fuzzy sets theory introduced a new concept, which means prepositions can take values from 0 to 1. This concept is called degrees of membership. Introduced by Zadeh in 1965 [55], the Fuzzy sets theory provides a mathematical tool capable of helping with decision taking in

an environment with imprecision variables, uncertainty and incomplete information.

A Fuzzy set can be defined as (S,f) where S is any set and f represents membership function. Every x element belongs to S, the $f(x)$ value defines the membership degree of x in the set(S,f). The x element is considered not included if $f(x) = 0$, totally included if $f(x) = 0$ and fuzzy member if $f(x) = 1$. An example of membership function is the Gaussiana, which is defined as:

$$f(x) = e^{\frac{-(x--m)^2}{2\sigma^2}} \tag{9}$$

where m is the mean and σ is the standard deviation of the S set.

According to Wu and Banzhaf [56], the Fuzzy logic is used to detect the anomalies in networks for two main reasons. The first one, the anomaly detection problems involve countless numeric attributes that are collected and derived statistically, which could cause a detection error. The second, the models that generate a normal profile of network behavior need to determine thresholds between the normal and anomalous behaviors. However, this interval is not well-defined and small changes (e.g., adversarial examples) on traffic behavior can cause false alarms. Considering these factors, the Fuzzy logic was used in this work to help with decision taking for anomaly detection.

2) ANOMALY DETECTION PHASE

The proposed model for anomaly detection in this work uses past traffic, the one predicted by LSTM and the Fuzzy logic. The first step is the ''fuzzification'' of sources for each one of the flow attributes being analyzed, applying the membership function. The membership function applied in this work is a modification of the Gaussian membership function, defined as:

$$f(y_t)_j = e^{\frac{-(x_t--y_t)^2}{2\hat{\sigma}_t^2}} \tag{10}$$

where x_t is the real traffic, y_t is the predicted traffic by LSTM and $\hat{\sigma}_t$ is the threshold generated by the Bienaymé-Chebyshev's inequality from the flow attribute j.

The Eq. 10 determines the membership degree of the normal traffic set. Therefore, to detect an anomaly we will apply its complement, defined as:

$$f_j' = 1 - f_j \tag{11}$$

The f_i' result represents the anomaly score of the flow attribute j. The anomaly scores are used to classify the traffic behavior to an instant data analysis. The process of ''defuzzification'' determines rather the traffic is ''normal'', ''Portscan'' or ''DDoS'', which are described in the following rules:

$$Rule \quad 1: \quad IF \quad \sum_{j=1}^{6} f_j' < \gamma \quad THEN \quad ''normal'' \tag{12}$$

$$Rule \quad 2: \quad IF \quad \sum_{j=1}^{6} f_j' \geq \gamma \quad AND \sum_{j=1}^{6} f_j' < \zeta \quad THEN \quad ''Portscan'' \tag{13}$$

$$Rule \quad 3: \quad IF \quad \sum_{j=1}^{6} f_j' \geq \zeta \quad THEN \quad ''DDoS'' \tag{14}$$

The values for γ and ζ scores were defined as 1.2362 and 3.3821, respectively. These values were rated by accuracy and are detailed on Section IV-C. Fig. 4 illustrates the anomaly score of all flow attributes during a day of network traffic analysis, which contains a DDoS and Portscan attack period. On the other hand, Fig. 5 illustrates the anomaly score sum of all six flow attributes. With the anomaly score calculated, the system can detect an attack based on the rules defined in (12), (13), and (14).

C. MITIGATION PHASE

The detection and identification of anomalies are essential stages that guaranty the operation and the services available throughout the network systems. After detecting an anomalous event, a mechanism must be used to minimize the effects caused by that event. The usual process to determine the effects caused by attacks is by mitigating, applying autonomic policies without the need for human interference, and aiming to ensure the network's resilience and operation. Thus, the proposed system consists of a module responsible for identifying the anomalous flows and mitigation policies are taken.

Mitigation policies are structured by using the **Event-Condition-Action** (ECA) model, which is considered adequate for the dynamic managing of policies. In this model, the **Event** refers to a specific anomaly and is associated with a set of rules. These rules are described as a set of **Conditions** that correspond to the context in which the anomaly took place. Finally, the **Action** is a countermeasure taken in relation to flows identified as malicious [57].

The main method used in applications for attack mitigation on SDN environments is to modify the flow table entry of the switch or to add a new flow entry. After detecting an attack, some characteristics must be identified, for example, source IP, IP destination, source port number, destination port number, and the kind of protocol. These characteristics help to identify the attacker and are fundamental for taking the countermeasures to minimize the damage an attack causes. A new entry on the flow table can be installed based on one or more of these characteristics, signaling that the packages that belong to the flow are from an attacker. Also, the actions taken can be the discard of these packets, anomalous traffic blockage and/or a honeypot redirection [58]

Based on the presented concepts, the mitigation module of the system is made by two policies to mitigate the detected anomalies. After the detection module's alarm goes off, the mitigation module takes action. The first step is to identify the suspect flows of the analysis interval. The identification of the suspected floes is made based on IP addresses analysis

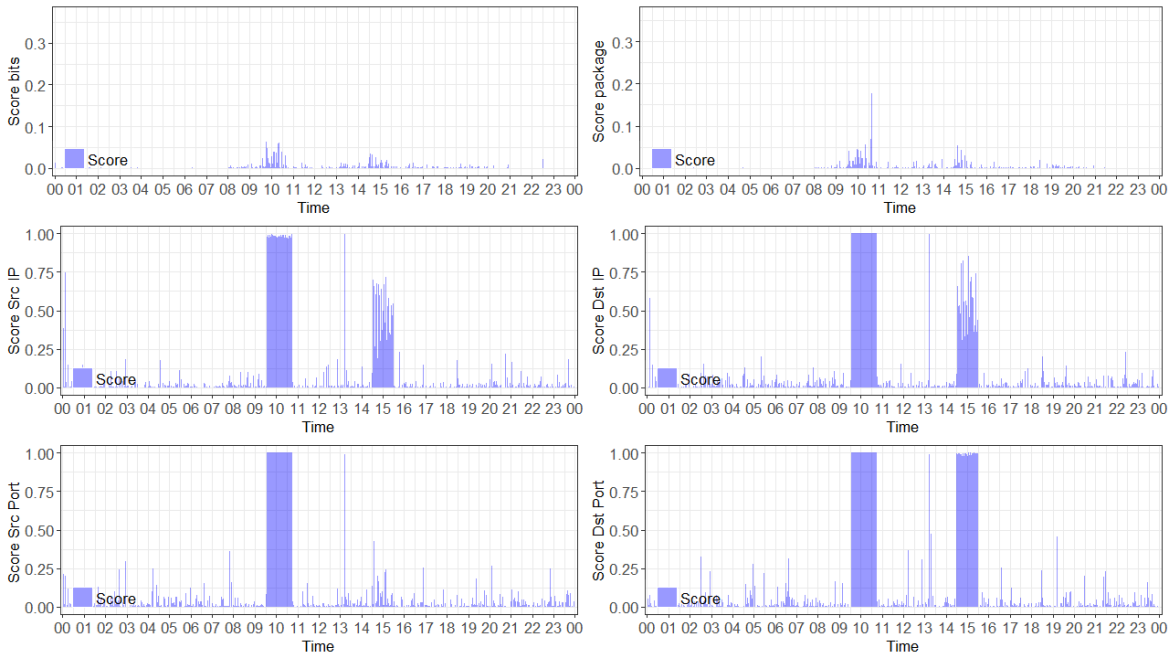

FIGURE 4. Anomaly score per flow attribute.

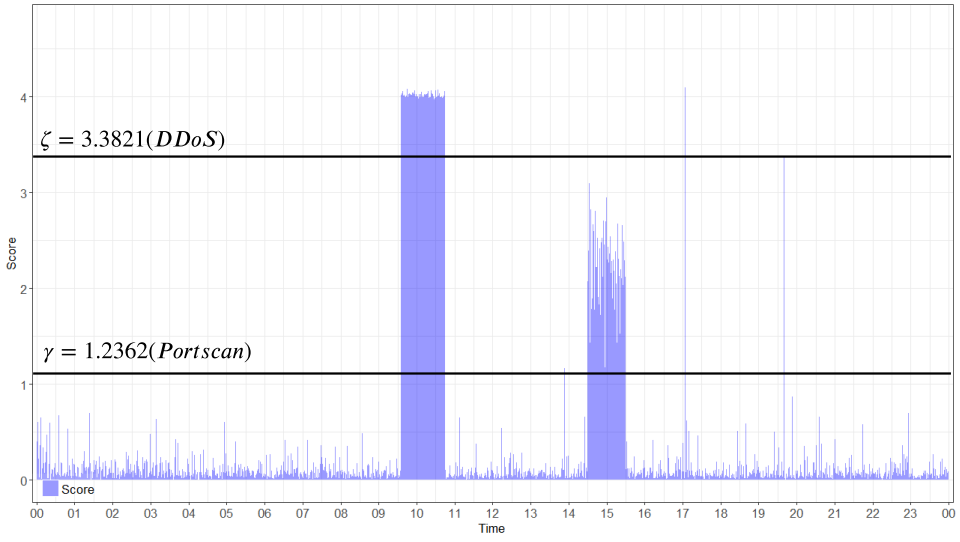

FIGURE 5. Anomaly score sum by six flows attributes.

and ports that make the anomalous interval. The ones that move toward the IP address that most receives flows are considered suspects.

By identifying the suspect flows, in case of an **Event** being launched by the detection module is a DDoS attack, a discard of the flows will be made based on the source IP addresses which appear more often on the suspected flows and which simultaneously have the same destination port. When launched **Event** is a Portscan attack, the process of identification of the attack is made by the origin IP address that presents the most variety of destination ports. This IP is considered an attacking one, and all its flows will be dropped. The process of mitigation is shown in the Algorithm 2.

There are anomalies that are not caused by malicious agents, but possess the same behavior of an attack. For example, a Flash crowd anomaly has the same characteristics of a DDoS attack, however, they are user performing legit requisitions. In the works of Giotis *et al.* [58] and Assis *et al.* [46], the authors suggest the implementation of a mechanism that maintains a list of IP flow attributes of legit users for a determined time of 5 minutes. This way, we also implemented a mechanism called *Safe List* that keeps a list of IP addresses

Algorithm 2 Mitigation Process

Require: Suspect flows
Ensure: Anomalous Packets Discard

```
if DDoS attack then
    Identify the destination IP address which receives the most flows
    Identify in those flows the attacker's IP address which have the same destination port
    if IPs and ports are on the Safe List then
        Forward packets
    else
        Drop packets
if Portscan attack then
    Identify the IP address that has received the most flows
    Identify in those flows the origin IP which presents the most variety of destination ports
    if IPs e ports are on the Safe List then
        Forward packets
    else
        Drop packets
```

and ports. So, this list is verified before starting the mitigation process.

IV. RESULTS AND DISCUSSION

The system was implemented by using the Python language and with development libraries for application of Deep Learning Keras and TensorFlow. The experiments were made in an environment with the following figures: Intel Core i5 2.21 GHz, 8 GB RAM and the operational system Windows 10. Default parameters as set, dropout = 0.2, loss function is MSE (Mean-Square Error), learning rate = 0.001, and optimizer was set as Adam proposed in [59], which is an adaptive learning rate optimization algorithm for training deep neural networks.

To demonstrate the effectiveness and efficiency of the system proposed, we applied tests applying from distinct scenarios. The test environment used in scenario 1 was a network topology with 120 hosts and the attacks were carried out in the periods of the day. In scenario 2 we used IP flows emulated from a public data called CICDDoS 2019 [60] from the Canadian Institute for Cybersecurity. This database contains different kinds of DDoS attacks and realistic traffic profiles.

A. SCENARIOS

The system performs a traffic behavior analysis each second. Therefore, the network flows must be collected in this time-lapse. Considering this analysis, in scenario 1, it was necessary to emulate the network behavior using the SDN Mininet network emulator [61], which allows the creation of realistic virtual networks consisting of controllers, hosts, links e switches on one single virtual machine. The Mininet uses light virtualization in the creation of personalized open code topologies and, it is broadly applied in this field to carry out researches and development of solutions for SDN environments. The experiments used a tool called Scapy [62] to inject traffic in the emulated network to make sure the emulated scenario is the closest it can get to a real SDN environment, with high rates of traffic going through the network.

TABLE 1. Information about the parameters of attacks in scenario 1.

Type of Attack	Attack Parameters
DDoS	Attackers: 16 Attacking IPs: 10.0.0.80 - 10.0.0.85 Victim IP address: 10.0.0.50 Time: 9:35 - 10:45
Portscan	Attacking IP: 10.0.0.10 Victim IP address: 10.0.0.24 Ports: 1 - 20000 Seconds to wait between packets: 0.15 Time: 14:30 - 15:30

Furthermore, to implement the anomaly detection and mitigation mechanism, we used the SDN controller Floodlight. A controller based on Java developed by BigSwich offers support to a wide variety of OpenFlow switches, virtual or physical, and it can cope with mixed networks, OpenFlow and no OpenFlow. The flows attributes used were: bit/s, packet/s, source IP entropy, destination IP entropy, source Port entropy and destination Port entropy.

Fig. 6 illustrates a topology emulated on scenario 1. The first scenario is formed by a topology in which its elements are distributed in the format of stars. This topology is made of central switches, in which six switches are connected. Each sub-network contains 20 hosts, totalizing 120 hosts. Two 24 hours day were emulated, that contains 86400 samples each day. The first day of emulation only contains samples of normal behavior of the network. This day was used in the LSTM training phase, as a semi-supervised training approach was used in its training. The second day of emulation was used to evaluate the system's operating performance in the detection and mitigation of attacks. Along with the emulation, two attacks were carried out with different intensities and duration time. There a DDoS attack and a Portscan attack. The information related to the parameters used in the attacks are shown in detail on TABLE 1. This dataset is available online.[1]

In scenario 2 we used the public dateset CICDDoS 2019 [60]. This set of data is distributed in two days, one for training and another for testing. The training set is made of 12 different kinds of DDoS attacks, being, NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebDDoS (ARME), SYN e TFTP. The second day, the testing day, contains 6 kinds of DDoS attacks, being NetBios, LDAP, MSSQL, UDP, UDP-Lag and SYN. The flows attributes used were the same as scenario 1.

[1] http://www.uel.br/grupos/orion/datasets.html

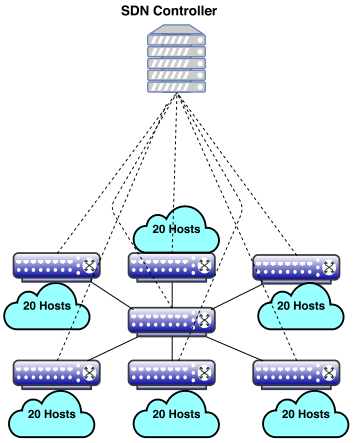

FIGURE 6. Network topology scenario 1 emulated on Mininet.

B. MÉTRICS AND TESTS

The tests applied aim to verify the efficiency of the suggested system, related to the modules that make it, detection and mitigation. The suggested module's performance results were analyzed using the following statistics metric [63]: precision, recall, false-positive rate.

1) precision: presents the percentage of intervals classified as anomalies, which are anomalies;
2) recall: measures how effective the model is in identifying the anomalous intervals about all the intervals;
3) false-positive rate: expresses a classification error, the traffic is identified as anomalous, but in fact, the traffic is normal.

These metrics can be easily calculated by the following equations:

$$\text{precision} = \frac{TP}{(TP + FP)} \tag{15}$$

$$\text{recall} = \frac{TP}{(TP + FN)} \tag{16}$$

$$\text{FPR} = \frac{FP}{(FP + FN)} \tag{17}$$

where, TP, TN, FP, FN mean true positive, true negative, false positive and false negative, respectively. Accuracy is a metric widely applied to anomaly detection works. However, it can lead to tendentious results where the dataset is unbalanced, which is the case of the data applied in this work. The dataset contains more normal samples than anomalous, and the system can classify all the samples correctly as normal and misclassify the anomalous samples, giving a tendentious result. The Precision metric can be used to solve this tendentious result and to emphasize the classification of correct anomalous samples.

The Receiver Operating Characteristics (ROC) [63] may be the combination of rates TP and FP, which gives a visual analysis of the system's capacity in detecting anomalous behaviors. However, to better quantify the efficiency between many classifiers, we analyze the area under the curve (AUC) of the ROC curve. The one with the highest value has the best ability to classify the samples. Therefore, AUC was applied to evaluate the proposed method with other models available in the literature.

TABLE 2. Contingency table (2 × 2).

	Positive test 2	Negative test 2	Sum row
Positive test 1	a	b	a+b
Negative test 1	c	d	c+d
Sum column	a+c	b+d	n

The efficiency of the module of mitigation was rated through the application of a statistic test called McNemar's Test, also through the dropped packet rate. The MacNemar Test a non-parametric test and its application is carried out through paired samples and nominal data. It is applied to contingency tables 2 × 2 with a dichotomous trace, which means, two behaviors (e.g., anomalous and normal) with the aim to verify if the marginal frequencies are equal or not [64] On TABLE 2 a generic example is illustrated of a contingency table 2 × 2 that presents the results of two tests in an sample of n individuals.

The null hypothesis indicates that the probabilities for each results are equal, that means, there was no change in the marginal frequencies and $p_a + p_b = p_a + p_c$ e $p_c + p_d = p_b + p_d$, where p_a, p_b, p_c, p_d indicate the theoretical probabilities of occurrences on the cells with the corresponding label. The null hypothesis and the alternative hypothesis are presented, respectively, as:

$$H_0 : p_b = p_c$$
$$H_1 : p_b \neq p_c$$

The MacNemar's test formula originated from the chi-square equation:

$$\chi^2 = \frac{(b-c)^2}{b+c} \tag{18}$$

χ^2 has a chi-squared distribution with a degree of freedom. If the result χ^2 is relevant, that means, that $p_b \neq p_c$ which means that the marginal frequencies are significantly different from one another the null hypothesis is rejected.

C. PARAMETERS EVALUATION

This section evaluates the results of the parameters used in the development of the proposed system. The first parameter to be looked into was the time step size used by the LSTM network in the traffic prediction phase. The values used for the test comprehend between 2 and 30 past samples of the traffic collected. The RMSE metric was used to determine the best time step size. The graphic present in Fig. 7 illustrates the values of RMSE obtained for each of the values evaluated. The time step size that presented the best result was equal to 5, with an RMSE value of 0.0445.

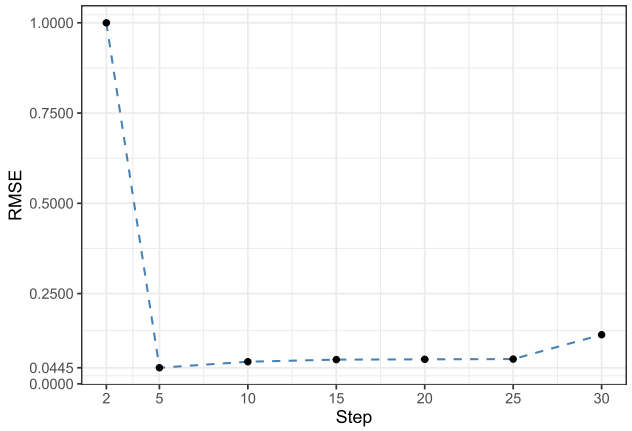

FIGURE 7. Time step size used for the LSTM traffic forecasting.

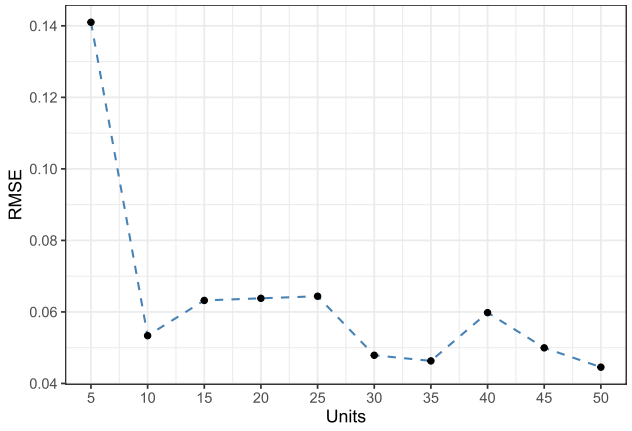

FIGURE 8. Number of units used for the LSTM traffic forecasting.

The next step was to define the number of hidden units. The evaluation of this parameter used the range of 5 to 100 units, and for each value the RMSE was evaluated. The number of units which presented the best result was 50 units, after which there was no significant improvement. In Fig. 8, we have a graphic with results obtained for each value of unit and its respective RMSE value.

The graphics present in Fig. 9 and Fig. 10 illustrate the evaluation of γ and ζ values to find the most adequate sum of anomaly score. The values were defined by varying γ and ζ and calculating its respective accuracies. The final value γ was defined with $argmax_{\gamma}(accuracy_{\gamma})$ and the ζ value was defined as $argmax_{\zeta}(accuracy_{\zeta})$. The score values for γ and ζ were defined as 1.2362 and 3.3821, respectively.

D. EVALUATION SCENARIO 1

To further validate our system, we compared our system with four other anomaly detection methods, which were applied to detect anomalies in SDN networks. The first method is the k-Nearest Neighbor (kNN) [65], a supervised classifier with a low computing cost, used to detect malicious events in a datacenter. The second method is the Multi-layer Perceptron (MLP) [66], an artificial neural network applied in the

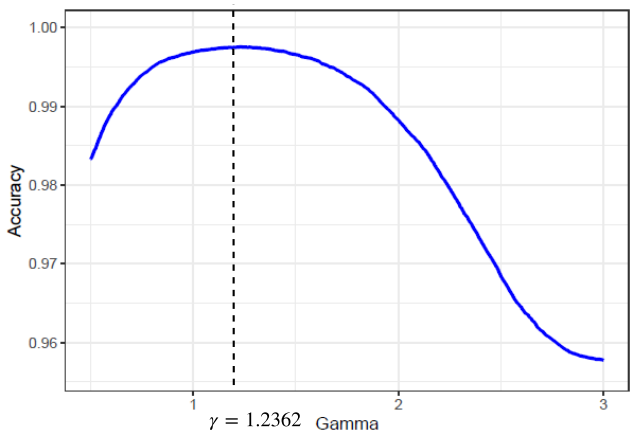

FIGURE 9. Accuracy evaluation for Gamma.

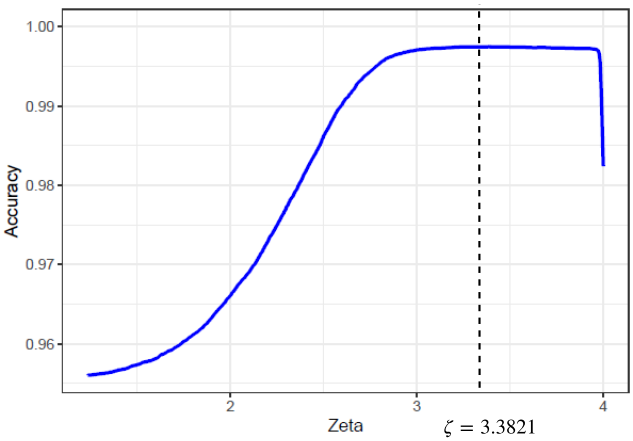

FIGURE 10. Accuracy evaluation for Zeta.

TABLE 3. Information about the samples for each class.

Type of Traffic	Number of Samples
Normal	78420
DDoS	4380
Portscan	3600
Total	86400

detection of DDoS attacks. Another method is based on Support Vector Machine (SVM) [67] to detect flooding attacks. We also compared it with LSTM-2 [68], which applied DL to detect DDoS attacks in the SDN environment. Finally, the recent method present in literature called Particle Swarm Optimization Digital Signature (PSO-DS) [46]. The heuristic method that used an unsupervised learning technique to detect DDoS and Portscan attacks on SDN networks.

To improve the comparison between the methods, on supervised approaches (kNN, SVM, MLP, and LSTM-2), we used a dataset for training that represents a day of network traffic data collection. This day is composed of normal traffic and by DDoS and Portscan attacks. The information for the number of samples to the classes are illustrated in Table 3.

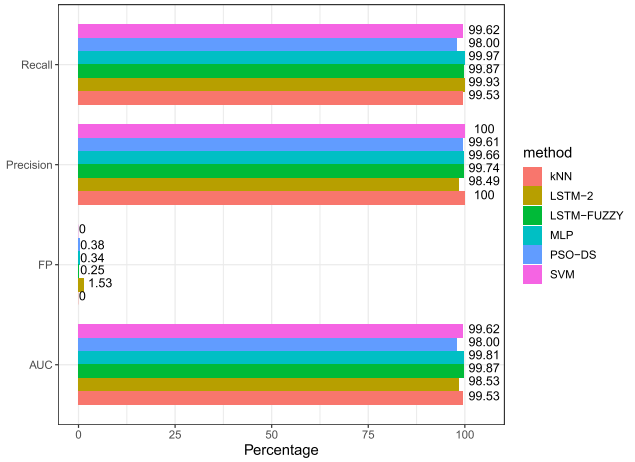

FIGURE 11. Detection results in the first scenario among LSTM-FUZZY and another methods.

A detailed analysis is illustrated in Fig. 11, where we present the metric results of compared methods. The LSTM-FUZZY presented a low false-positive rate, obtaining a value of 0.25%. The compared method LSTM-2 presented the highest false-positive rate, reaching 1.53%. On the other hand, the SVM and kNN methods didn't present false-positive rates. Regarding the recall and precision metrics, all the methods presented values superior to 98%. None of the methods reached better performance in all the metrics evaluated.

Fig. 12 presents the ROC curves, a visual comparison between the compared methods. Through the ROC curve, it is possible to determine which of the methods present the most adequate aptitude to detect anomalies. By analyzing the obtained results, it is clear that the LSTM-FUZZY approach obtained the best results among the other compared methods. The LSTM-FUZZY presented an AUC value of 99.87%, implying that the method presented the higher true positive rate with the lowest false-positive rate.

TABLE 4. Contingency table to evaluate the mitigation process on scenario 1.

	normal (after)	anomalous (after)
normal (before)	78323	86
anomalous (before)	7728	262

1) MITIGATION

From the alarms generated by the classification process of LSTM-FUZZY, mitigation policies were applied. Fig. 13 presents the traffic attributes in green without the application of mitigation and in blue is the traffic after the mitigation process. In the period between 9:45:00 and 10:35:00, we have a DDoS attack report, in this period we can see the increase of the packet and bits rate when the mitigation module is disabled, but by activating the module the traffic tends to go back to its normality due to discards of anomalous packets. The period of the Portscan attack between 14:30 and 15:30 causes minor changes in the traffic behavior, with the application of mitigation the affected attributes also go back to its normality.

In this scenario, the mitigation analysis through the McNemar's test and dropped packet rates were also applied. The significance level for the McNemar's test was $\alpha = 5\%$. The TABLE 4 offers information on the traffic classification between anomalous and normal before and after the mitigation process. By applying the test to the information on the table, the p-value results were lower than $2.2 \times e^{-16}$ that is smaller than the α value. Therefore, the null hypothesis in this scenario was also rejected. It indicates that there was indeed a difference in the frequencies. Thus, the mitigation was efficient in minimizing the threat effects. Also, the rate of anomalous packages dropped by the system was 99.88%. This result shows that almost all the anomalous packets were dropped.

E. EVALUATION SCENARIO 2

This scenario aims to evaluate the module of system detection, applying different kinds of DDoS attacks. As mentioned

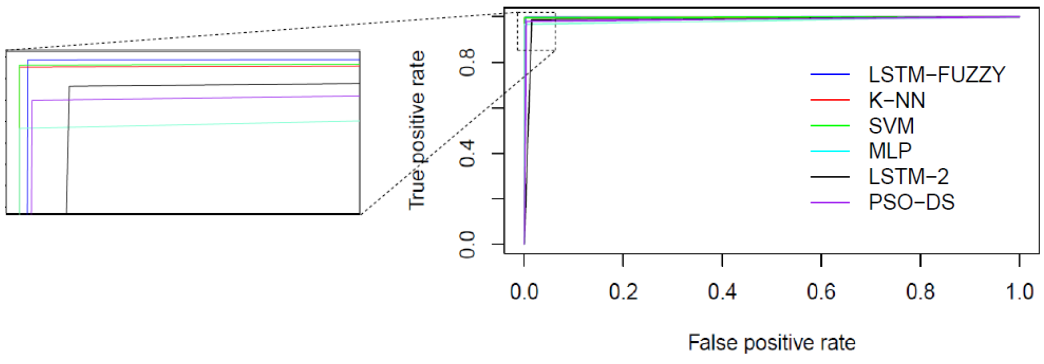

FIGURE 12. ROC curves of the methods compared scenario 1.

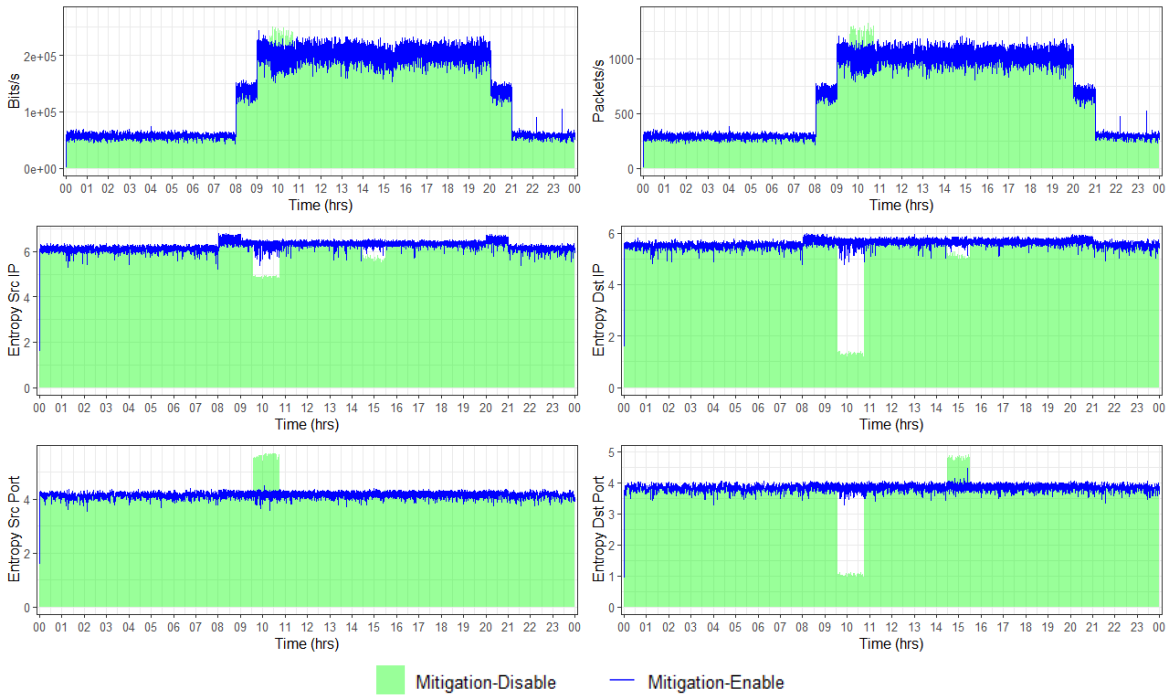

FIGURE 13. Graph showing non-mitigated traffic and mitigated traffic for intervals with anomaly on scenario 1.

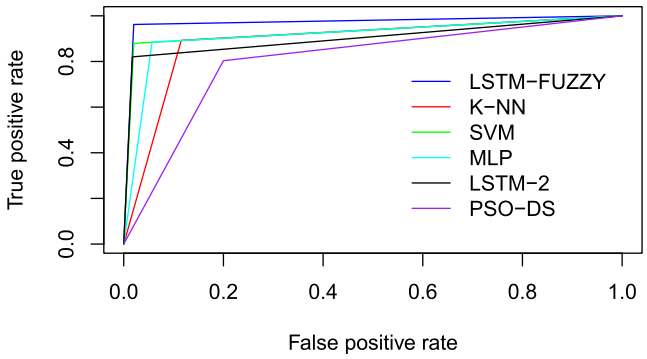

FIGURE 14. ROC curves of the methods compared scenario 2.

previously, the CICDDoS 2019 dataset [60], developed by the Canadian Institute for Cybersecurity, is made of two days (train and test). The training day is made of 12 kinds of DDoS attacks, and the test day contains six kinds of DDoS attacks.

As mentioned, the system suggested in this work does the analysis of traffic every 1 second. Thus, it was necessary to run a pre-process of the CICDDoS 2019 dataset to summarize the flows into groups of one-second intervals based on their timestamp feature. After grouping, we noted that all the intervals were made by only anomalous samples. To solve this problem, we separated the flows by anomalous and normal before the process of grouping in 1 second intervals.

However, the size of the flows samples containing DDoS attacks is superior to the normal data due to the characteristics of the attacks. It is not a problem to the LSTM-FUZZY, because on the training stage, the method only uses the normal samples to characterize the traffic, but it can generate overfitting for the methods SVM, kNN, MLP, and LSTM-2 that during the training a supervised approach is applied. To retain the characteristics and the representative of the applied data in the training, the solution applied to solve this problem was to sample the flows randomly for each kind of attack. Through empiric tests, for each kind of attack we selected a proportion of 5 times the normal flows. The set of training was reduced but it maintained enough sample quantity for the training process.

As executed in the first scenario, the efficiency of LSTM-FUZZY was compared to classic methods, kNN, SVM, MLP, LSTM-2, and PSO-DS. Fig. 15 illustrates the results of metrics obtained for each one of them. About the recall metric, it is clear that the LSTM-FUZZY obtained a performance superior to the other compared methods, obtaining a value of 93.13% for this metric, followed by LSTM-2, PSO-DS, kNN, MLP and SVM, which reached the rates of 90.53%, 89.66%, 89.27%, 87.92%, and 87.92%, respectively.

The next evaluated metric was the precision one, the LSTM-FUZZY again reached the best result, with a rate of 97.89%, the remaining ones were SVM, LSTM-2, MLP, kNN, and PSO-DS, reaching rates of 97.74%, 96.61%, 94.98%, 89.27%, and 81.19%, respectively. On the other hand, in comparison to the false-positive rate, the LSTM-FUZZY and the SVM reached the same rate of 2.2%, which

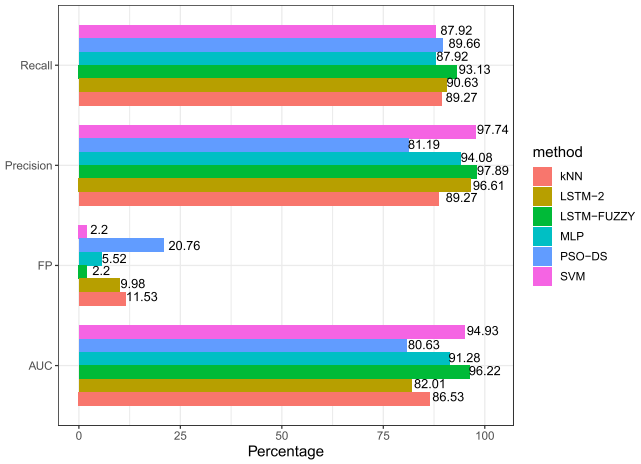

FIGURE 15. Detection results in the second scenario among LSTM-FUZZY and others methods.

could be considered a good result. After it was the MLP, LSTM-2, kNN and the PSO-DS, with retrospective values of 5.52%, 9.98%, 11.53%, and 20.76%. The proposed system showed superior results to the other compared methods, except for the SVM that obtained similar results. However, when using the ROC curve it was possible to observe the improvement between the compared methods more clearly. Despite the similar results, the performance applied by the proposed system is a significant improvement, as current computer networks operate with links with high transmission rates. Over a day of network operation, a small percentage of undetected attacks could cause damage to its operation. For instance, in October 2016, a DDoS attack with 100 thousand malicious endpoints surpassed a bandwidth of 1.2 Tbps [69]. As the outcomes presented in the first scenario, the LSTM-FUZZY method also fared better on the average than the other compared methods on the second scenario, achieving promising test outcomes that make it an efficient technique on detecting different kinds of DDoS attacks.

Just as in the previous scenario, the ROC curve was used to determine which method presented the best performance in detecting attacks. Fig. 14 presents the visual analysis of the ROC curve. Through AUC, we can see that the LSTM-FUZZY was the one that reached the best balance between the true-positive rates and the false-positive rates, reaching a value of 96.22%. Followed by the SVM, MLP, kNN, LSTM-2, and PSO-DS with the following values 94.93%, 91.28%, 86.53%, 82.01%, and 80.63.

1) MITIGATION

In this scenario, we evaluated the efficiency to mitigate the DDoS attacks from CICDDoS 2019 dataset. Fig. 16 presents the traffic behavior from the test day where there is the DDoS attack report with the mitigation module deactivated and compares its behavior when the mitigation is activated. The traffic generated without the application of mitigation policy is represented in the green area, and the blue line shows the traffic after the application of mitigation against the DDoS attacks. Visually it is possible to see when the attacks are

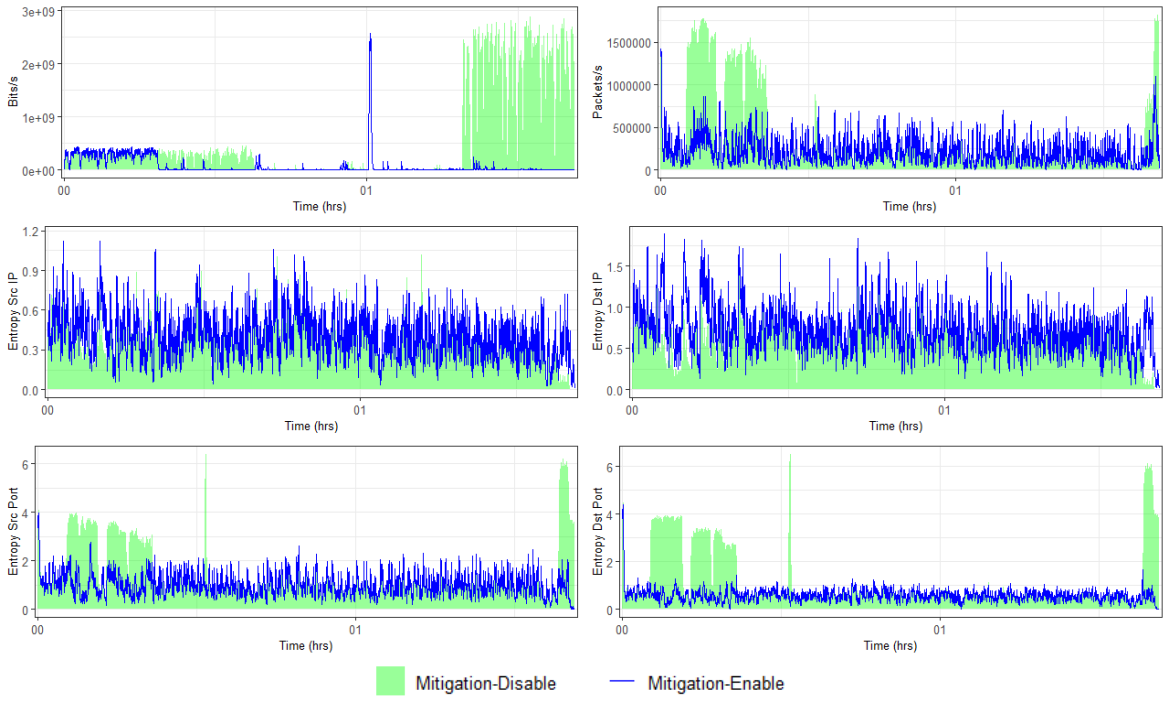

FIGURE 16. Traffic analysis with mitigation module disable and enable on test day from CICDDoS 2019 dataset.

TABLE 5. Contingency table applied for evaluation of mitigation on scenario 2.

	normal (after)	anomalous (after)
normal (before)	4811	136
anomalous (before)	2175	68

mitigated, the attribute values being analyzed return to its expected behavior.

The MacNemar's test was applied with a level of significance of $\alpha = 5\%$ and the null hypothesis that the marginal frequencies are equal. After applying the test on the contingency TABLE 5, the p-value result was $2.2 \times e^{-16}$ that is lower than the value α. Thus, the null hypothesis is rejected, which indicates that there was a difference in the marginal frequencies, and the mitigation was effective. Moreover, the anomalous packets rates discarded was 99.20%, which implies that the majority of the anomalous packages were mitigated.

V. CONCLUSION

In this work, we presented a modular system for detection and mitigation of anomalies in SDN networks. The system is made of three modules where its activities are carried out in an automatized way to make monitoring, detection, and mitigation of attacks easier. In the first module, responsible for the characterization of traffic, we developed a new approach to predict the normal behavior of the network operation, applying an approach of Long Short-Term Memory (LSTM) semi-supervised using IP flows. In the second module, we proposed a mechanism to recognize attacks, through the application of Bienaymé-Chebyshev's inequality along with the Fuzzy logic. Finally, in the third module, we applied automatized mitigation policies to minimize the damage caused by attacks and to maintain the requirement of network operation.

To validate the development system, we employed two scenarios with distinct characteristics. In the first scenario, we used emulated SDN data, using the Mininet emulator and the Floodlight controller, containing periods of DDoS and Portscan attacks. In the second scenario, we used a public dataset called CICDDoS 2019. This dataset is made of 12 kinds of different DDoS attacks. To test the detection module, we compared the LSTM-FUZZY with the other methods present in the literature, SVM, kNN, MLP, LSTM-2, and PSO-DS. In both scenarios we compared the performance between the suggested method and the others. According to the results presented, the LSTM-FUZZY presented a superior performance compared to the others, reaching a low false-positive rate and high precision, recall, and AUC rates.

In the first scenario, we applied mitigation policies based on the kind of attack identified by the detection module. In this module, we identified the suspect flows, based on the analysis of IP addresses and ports that make the anomalous interval. The flows identified as suspects were dropped. Through the McNemar's test and dropped anomalous packets rate, it was shown that the module obtained a satisfactory performance, minimizing the effects of the attacks.

The LSTM power to learn to extract short and long-term patterns allowed the application to predict the normal behavior of the network traffic. The module produced adequate predictions close to real traffic behavior, and it was possible to apply them in the detection stage. The Fuzzy Logic characteristics allowed anomaly detection in an unsupervised way, implying that the system does not need labeled data. The advantage of using this technique makes the system operation easier and discards the need to use a labeled dataset, which demands much work and could be full of human errors. Moreover, the Fuzzy Logic acts on the detection of different DDoS attacks with a low false-positive rate, allowing the system to act on the present SDN environment with high accuracy to detect and low false alarms.

The results obtained show that the modules that made the proposed system were efficient, meeting the goals assigned to each one of them. The execution of the activities carried out by the system is automatic, which means the process of monitoring, identification of adverse events, and the countermeasures are carried out without the need for human interference. The monitoring and managing of the network is a complex activity. The application of an autonomous system helps the assigned tasks to the administrator to maintain and guaranty the network's operation to its fullest. Hence, the system developed in this work can be applied to collaborate and facilitate management procedures and to guaranty the availability of the services offered.

The modular architecture of the system allows the maintenance and adaptation of other techniques to characterize traffic, detection, and mitigation of anomalies in SDN environments. This characteristic allows the adaptation of the system as the network dynamics change, and new security demands emerge. Thus, future works can explore other vulnerabilities and incorporate mitigation policies to meet new demands that might emerge in SDN network environments. Another point that could be extended is the exploration of more tests in other scenarios with different types of topology and attacks.

REFERENCES

[1] R. Khan, P. Kumar, D. N. K. Jayakody, and M. Liyanage, "A survey on security and privacy of 5G technologies: Potential solutions, recent advancements, and future directions," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 196–248, 1st Quart., 2020.

[2] A. Rego, L. Garcia, S. Sendra, and J. Lloret, "Software defined network-based control system for an efficient traffic management for emergency situations in smart cities," *Future Gener. Comput. Syst.*, vol. 88, pp. 243–253, Nov. 2018.

[3] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges," *Comput. Netw.*, vol. 167, Feb. 2020, Art. no. 106984.

[4] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[5] T. Das, V. Sridharan, and M. Gurusamy, "A survey on controller placement in SDN," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 472–503, 1st Quart., 2020.

[6] O. Salman, I. Elhajj, A. Chehab, and A. Kayssi, "IoT survey: An SDN and fog computing perspective," *Comput. Netw.*, vol. 143, pp. 221–246, Oct. 2018.

[7] M. B. M. Noor and W. H. Hassan, "Current research on Internet of Things (IoT) security: A survey," *Comput. Netw.*, vol. 148, pp. 283–294, Jan. 2019.
[8] Z. Li, W. Xing, S. Khamaiseh, and D. Xu, "Detecting saturation attacks based on self-similarity of OpenFlow traffic," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 1, pp. 607–621, Mar. 2020.
[9] M. Jonker, A. King, J. Krupp, C. Rossow, A. Sperotto, and A. Dainotti, "Millions of targets under attack: A macroscopic characterization of the DoS ecosystem," in *Proc. Internet Meas. Conf.*, New York, NY, USA, Nov. 2017, pp. 100–113.
[10] T. A. Pascoal, I. E. Fonseca, and V. Nigam, "Slow denial-of-service attacks on software defined networks," *Comput. Netw.*, vol. 173, May 2020, Art. no. 107223.
[11] V. Varadharajan, K. Karmakar, U. Tupakula, and M. Hitchens, "A policy-based security architecture for software-defined networks," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 4, pp. 897–912, Apr. 2019.
[12] V. Varadharajan and U. Tupakula, "Counteracting attacks from malicious end hosts in software defined networks," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 1, pp. 160–174, Mar. 2020.
[13] C. Yoon, T. Park, S. Lee, H. Kang, S. Shin, and Z. Zhang, "Enabling security functions with SDN: A feasibility study," *Comput. Netw.*, vol. 85, pp. 19–35, Jul. 2015.
[14] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 602–622, 1st Quart., 2016.
[15] K. Kalkan, G. Gur, and F. Alagoz, "Defense mechanisms against DDoS attacks in SDN environment," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 175–179, Sep. 2017.
[16] C. Zhang, G. Hu, G. Chen, A. K. Sangaiah, P. Zhang, X. Yan, and W. Jiang, "Towards a SDN-based integrated architecture for mitigating IP spoofing attack," *IEEE Access*, vol. 6, pp. 22764–22777, 2018.
[17] N. Moustafa, J. Hu, and J. Slay, "A holistic review of network anomaly detection systems: A comprehensive survey," *J. Netw. Comput. Appl.*, vol. 128, pp. 33–55, Feb. 2019.
[18] S. Mahdavifar and A. A. Ghorbani, "Application of deep learning to cybersecurity: A survey," *Neurocomputing*, vol. 347, pp. 149–176, Jun. 2019.
[19] W. Wu, R. Li, G. Xie, J. An, Y. Bai, J. Zhou, and K. Li, "A survey of intrusion detection for in-vehicle networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 919–933, Mar. 2020.
[20] M. L. Proenca, B. B. Zarpelao, and L. S. Mendes, "Anomaly detection for network servers using digital signature of network segment," in *Proc. Adv. Ind. Conf. Telecommun./Service Assurance Partial Intermittent Resour. Conf./E-Learn. Telecommun. Workshop (AICT/SAPIR/ELETE)*, Jul. 2005, pp. 290–295.
[21] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, "A detailed investigation and analysis of using machine learning techniques for intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 686–728, 1st Quart., 2019.
[22] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for IoT security based on learning techniques," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2671–2701, 3rd Quart., 2019.
[23] R. K. Malaiya, D. Kwon, S. C. Suh, H. Kim, I. Kim, and J. Kim, "An empirical evaluation of deep learning for network anomaly detection," *IEEE Access*, vol. 7, pp. 140806–140817, 2019.
[24] Y. Zeng, H. Gu, W. Wei, and Y. Guo, "*Deep − full − range*: A deep learning based network encrypted traffic classification and intrusion detection framework," *IEEE Access*, vol. 7, pp. 45182–45190, 2019.
[25] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018.
[26] A. Aldweesh, A. Derhab, and A. Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues," *Knowl.-Based Syst.*, vol. 189, Feb. 2020, Art. no. 105124.
[27] F. A. Khan, A. Gumaei, A. Derhab, and A. Hussain, "TSDL: A two-stage deep learning model for efficient network intrusion detection," *IEEE Access*, vol. 7, pp. 30373–30385, 2019.
[28] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.
[29] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *J. Inf. Secur. Appl.*, vol. 50, Feb. 2020, Art. no. 102419.
[30] C. Gkountis, M. Taha, J. Lloret, and G. Kambourakis, "Lightweight algorithm for protecting SDN controller against DDoS attacks," in *Proc. 10th IFIP Wireless Mobile Netw. Conf. (WMNC)*, Sep. 2017, pp. 1–6.
[31] A. AlEroud and I. Alsmadi, "Identifying cyber-attacks on software defined networks: An inference-based intrusion detection approach," *J. Netw. Comput. Appl.*, vol. 80, pp. 152–164, Feb. 2017.
[32] G. Fernandes, J. J. P. C. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, and M. L. Proença, "A comprehensive survey on network anomaly detection," *Telecommun. Syst.*, vol. 70, no. 3, pp. 447–489, Mar. 2019.
[33] A. S. Da Silva, J. A. Wickboldt, L. Z. Granville, and A. Schaeffer-Filho, "ATLANTIC: A framework for anomaly traffic detection, classification, and mitigation in SDN," in *Proc. NOMS-IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2016, pp. 27–35.
[34] G. Garg and R. Garg, "Detecting anomalies efficiently in SDN using adaptive mechanism," in *Proc. 5th Int. Conf. Adv. Comput. Commun. Technol.*, Feb. 2015, pp. 367–370.
[35] S. M. Mousavi and M. St-Hilaire, "Early detection of DDoS attacks against SDN controllers," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2015, pp. 77–81.
[36] L. F. Carvalho, T. Abrão, L. D. S. Mendes, and M. L. Proença, "An ecosystem for anomaly detection and mitigation in software-defined networking," *Expert Syst. Appl.*, vol. 104, pp. 121–133, Aug. 2018.
[37] A. H. Hamamoto, L. F. Carvalho, L. D. H. Sampaio, T. Abrao, and M. Proença, "Network anomaly detection system using genetic algorithm and fuzzy logic," *Expert Syst. Appl.*, vol. 92, pp. 390–402, Feb. 2018.
[38] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 41–50, Feb. 2018.
[39] P. K. Roy, J. P. Singh, and S. Banerjee, "Deep learning to filter SMS spam," *Future Gener. Comput. Syst.*, vol. 102, pp. 524–533, Jan. 2020.
[40] X. Yuan, C. Li, and X. Li, "DeepDefense: Identifying DDoS attack via deep learning," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, May 2017, pp. 1–8.
[41] C. Li, Y. Wu, X. Yuan, Z. Sun, W. Wang, X. Li, and L. Gong, "Detection and defense of DDoS attack-based on deep learning in OpenFlow-based SDN," *Int. J. Commun. Syst.*, vol. 31, no. 5, p. e3497, Mar. 2018.
[42] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for network intrusion detection in software defined networking," in *Proc. Int. Conf. Wireless Netw. Mobile Commun. (WINCOM)*, Oct. 2016, pp. 258–263.
[43] S. K. Dey and M. M. Rahman, "Flow based anomaly detection in software defined networking: A deep learning approach with feature selection method," in *Proc. 4th Int. Conf. Electr. Eng. Inf. Commun. Technol. (iCEE-iCT)*, Sep. 2018, pp. 630–635.
[44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[45] E. H. M. Pena, S. Barbon, J. J. P. C. Rodrigues, and M. L. Proenca, "Anomaly detection using digital signature of network segment with adaptive ARIMA model and paraconsistent logic," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2014, pp. 1–6.
[46] M. V. O. De Assis, M. P. Novaes, C. B. Zerbini, L. F. Carvalho, T. Abrao, and M. L. Proenca, "Fast defense system against attacks in software defined networks," *IEEE Access*, vol. 6, pp. 69620–69639, 2018.
[47] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948.
[48] B. Yang, S. Sun, J. Li, X. Lin, and Y. Tian, "Traffic flow prediction using LSTM with feature enhancement," *Neurocomputing*, vol. 332, pp. 320–327, Mar. 2019.
[49] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," in *Proc. 41st Int. ACM SIGIR Conf. Res. Development Inf. Retr.*, New York, NY, USA, 2018, pp. 95–104.
[50] J. Bhatia, R. Dave, H. Bhayani, S. Tanwar, and A. Nayyar, "SDN-based real-time urban traffic analysis in VANET environment," *Comput. Commun.*, vol. 149, pp. 162–175, Jan. 2020.
[51] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, "DeepAnT: A deep learning approach for unsupervised anomaly detection in time series," *IEEE Access*, vol. 7, pp. 1991–2005, 2019.
[52] X. Qing and Y. Niu, "Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM," *Energy*, vol. 148, pp. 461–468, Apr. 2018.

[53] B. G. Amidan, T. A. Ferryman, and S. K. Cooley, "Data outlier detection using the Chebyshev theorem," in *Proc. IEEE Aerosp. Conf.*, Mar. 2005, pp. 3–8.
[54] C. Taylor and J. Alves-Foss, "An empirical analysis of NATE: Network analysis of anomalous traffic events," in *Proc. Workshop New Secur. Paradigms*, New York, NY, USA, 2002, pp. 18–26.
[55] U. R. Rosyara, D. Vromman, and E. Duveiller, "Canopy temperature depression as an indication of correlative measure of spot blotch resistance and heat stress tolerance in spring wheat," *J. Plant Pathol.*, vol. 90, no. 1, pp. 103–107, 2008.
[56] S. X. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A review," *Appl. Soft Comput.*, vol. 10, no. 1, pp. 1–35, Jan. 2010.
[57] R. Sahay, G. Blanc, Z. Zhang, K. Toumi, and H. Debar, "Adaptive policy-driven attack mitigation in SDN," in *Proc. 1st Int. Workshop Secur. Dependability Multi-Domain Infrastruct. (XDOMO)*, New York, NY, USA, 2017, pp. 4:1–4:6.
[58] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments," *Comput. Netw.*, vol. 62, pp. 122–136, Apr. 2014.
[59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, Y. Bengio and Y. LeCun, Eds. San Diego, CA, USA, May 2015.
[60] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *Proc. Int. Carnahan Conf. Secur. Technol. (ICCST)*, Oct. 2019, pp. 1–8.
[61] (2019). Mininet Team. *Mininet Overview*. Accessed: Dec. 3, 2019. [Online]. Available: http://mininet.org/overview/
[62] P. Biondi. (2019). *Scapy*. Accessed: Dec. 3, 2019. [Online]. Available: http://www.secdev.org/projects/scapy/
[63] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.
[64] X. Sun and Z. Yang, "Generalized McNemar's test for homogeneity of the marginal distributions," in *Proc. SAS Global Forum*, vol. 382, 2008, pp. 1–10.
[65] P. Xiao, W. Qu, H. Qi, and Z. Li, "Detecting DDoS attacks against data center with correlation analysis," *Comput. Commun.*, vol. 67, pp. 66–74, Aug. 2015.
[66] M. Wang, Y. Lu, and J. Qin, "A dynamic MLP-based DDoS attack detection method using feature selection and feedback," *Comput. Secur.*, vol. 88, Jan. 2020, Art. no. 101645.
[67] T. V. Phan, T. Van Toan, D. Van Tuyen, T. T. Huong, and N. H. Thanh, "OpenFlowSIA: An optimized protection scheme for software-defined networks from flooding attacks," in *Proc. IEEE 6th Int. Conf. Commun. Electron. (ICCE)*, Jul. 2016, pp. 13–18.
[68] R. Priyadarshini and R. K. Barik, "A deep learning based intelligent framework to mitigate DDoS attack in fog environment," *J. King Saud Univ.-Comput. Inf. Sci.*, early access, Apr. 24, 2019, doi: 10.1016/j.jksuci.2019.04.010.
[69] E. Viegas, A. Santin, A. Bessani, and N. Neves, "BigFlow: Real-time and reliable anomaly-based intrusion detection for high-speed networks," *Future Gener. Comput. Syst.*, vol. 93, pp. 473–485, Apr. 2019.

MATHEUS P. NOVAES received the master's degree in computer science from the State University of Londrina (UEL), Brazil, where he is currently pursuing the Ph.D. degree with the Electrical Engineering Department. He has been a member of the Research Group Computer Networks and Data Communication, Computer Science Department, UEL. His research interest includes management and security of computer networks.

LUIZ F. CARVALHO received the master's degree in computer science from the State University of Londrina, in 2014, and the Ph.D. degree in electrical engineering and telecommunications from the State University of Campinas, in 2018. He has experience in computer science with emphasis in computer networks and is part of the Research Group Computer Networks and Data Communication. His main research interests include management and security of computer networks and software-defined networks.

JAIME LLORET (Senior Member, IEEE) received the M.Sc. degrees in physics and electronic engineering from the University of Valencia, Valencia, Spain, in 1997 and 2003, respectively, and the Ph.D. degree in telecommunication engineering (Dr.Ing.) from the Polytechnic University of Valencia, Valencia, in 2006.

He is currently an Associate Professor with the Department of Communications, Polytechnic University of Valencia.

Dr. Lloret was the Internet Technical Committee Chair, from 2014 to 2015. He is also the Chair of IEEE 1907.1. He is also the Director of the Research Institute Integrated Management Coastal Research Institute (IGIC) and the Head of the Innovation Group Active and collaborative techniques and use of technologic resources in the education (EITACURTE). He has been the General Chair of 36 international workshops and conferences. He is also the Co-Editor-in-Chief of *Ad Hoc and Sensor Wireless Networks* and the Editor-in-Chief of *Network Protocols and Algorithms*.

MARIO LEMES PROENÇA, Jr. received the M.Sc. degree in computer science from the Informatics Institute, Federal University of Rio Grande do Sul (UFRGS), in 1998, and the Ph.D. degree in electrical engineering and telecommunications from the State University of Campinas (UNICAMP), in 2005. He is currently an Associate Professor and the Leader of the Research Group that studies computer networks in the Computer Science Department, State University of Londrina (UEL), Brazil. He has authored or coauthored over 100 articles in refereed international journals and conferences, books chapters, and one software register patent. His research interests include computer networks, network operations, management and security, and IT governance. He has supervised 14 M.Sc. and three Ph.D. students. He has been a Master's Supervisor in computer science with the State University of Londrina and a Ph.D. Supervisor with the Department of Electrical Engineering, UEL.