Computers and Electrical Engineering 86 (2020) 106738

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Near real-time security system applied to SDN environments in IoT networks using convolutional neural network☆

Marcos V.O. de Assis[a], Luiz F. Carvalho[b], Joel J.P.C. Rodrigues[c,d], Jaime Lloret[e,*], Mario L. Proença Jr[f]

[a] Engineering and Exact Department, Federal University of Paraná (UFPR), Paraná, Brazil
[b] Federal University of Technology - Paraná (UTFPR), Apucarana, Brazil
[c] Federal University of Piauí, Teresina - PI, Brazil
[d] Instituto de Telecomunicações, Portugal
[e] Integrated Management Coastal Research Institute, Universitat Politecnica de Valencia, Valencia, Spain
[f] Computer Science Department, State University of Londrina (UEL), Paraná, 86057-970, Brazil

ARTICLE INFO

Article history:
Received 3 November 2019
Revised 22 June 2020
Accepted 23 June 2020

Keywords:
Software-defined Network
Internet of Things
DDoS
CNN
Botnet
Deep Learning

ABSTRACT

The Internet of Things (IoT) paradigm brings new and promising possibilities for services and products. The heterogeneity of IoT devices highlights the inefficiency of traditional networks' structures to support their specific requirements due to their lack of flexibility. Thus, Software-defined Networking (SDN) is commonly associated with IoT since this architecture provides a more flexible and manageable network environment. As shown by recent events, IoT devices may be used for large scale Distributed Denial of Service (DDoS) attacks due to their lack of security. This kind of attack is commonly detected and mitigated at the destination-end network but, due to the massive volume of information that IoT botnets generate, this approach is becoming impracticable. We propose in this paper a near real-time SDN security system that both prevents DDoS attacks on the source-end network and protects the sources SDN controller against traffic impairment. For this, we apply and test a Convolutional Neural Network (CNN) for DDoS detection, and describe how the system could mitigate the detected attacks. The performance outcomes were performed in two test scenarios, and the results pointed out that the proposed SDN security system is promising against next-generation DDoS attacks.

1. Introduction

Internet of Things (IoT) is a paradigm defined by Tudosa et al. [1] as an evolving technology where every device can be both connected through a network and controllable from a remote station. It envisions a world where a significant amount of everyday objects communicate through wired and wireless networks [2]. The amount of information traveling over the Internet has been growing exponentially in past years, mainly due to the popularization of cloud computing solutions and connected applications, such as video conferences, IP video surveillance systems and so on. According to Chaabouni et al.

☆ This paper is for regular issues of CAEE. Reviews processed and recommended for publication to the Editor-in-Chief by Area Editor Dr. Huimin Lu.

* Corresponding author.
E-mail addresses: marcos.assis@ufpr.br (M.V.O. de Assis), luizfcarvalho@utfpr.edu.br (L.F. Carvalho), joeljr@ieee.org (J.J.P.C. Rodrigues), jlloret@dcom.upv.es (J. Lloret), proenca@uel.br (M.L. Proença Jr).

https://doi.org/10.1016/j.compeleceng.2020.106738
0045-7906/

[3], the IoT market is rapidly growing, starting with 2 billion devices by 2006 to a projection of 200 billion by 2020 (a 200% rise). With recent advances in communication technologies, the IoT is gaining visibility among researchers [1,4].

One of the main issues relating to IoT networks is its heterogeneous characteristic, since different applications have specific network requirements to optimize the system's operation. In [5] the authors highlight some examples of this IoT characteristic: in smart vehicles applications, the information exchange would require almost zero latency; in industrial sensor networks, besides the low latency, a minimal packet loss would be required; in mobile video surveillance network, the latency and packet loss are not critical, but would require a higher bandwidth. According to Caraguay et al. [6], these specific network requirements are incompatible with the traditional networking model, which have limitations regarding scalability, mobility and amount of traffic. Thus, traditional networks are inefficient to satisfy the new requirements of IoT environments.

A new paradigm that aims to provide scalability and flexibility to network's management process is the Software-defined networking (SDN). SDN enables centralized network management, allowing efficient configuration and optimization by transforming the traditional "black-box" network components into "white-box" software-controlled ones. This abstraction is possible by decoupling the control and data planes, where all control functions are implemented in a programmable central controller. This controller, in turn, sends packet forwarding and management policies to SDN controlled switches and routers, dynamically coordinating their operation and, consequently, the network behavior. These features make SDN a promising environment for the development and operation of IoT solutions [7].

Besides the advantages of the services provided by IoT, we recently witnessed its side effects relating to network security. According to Kim et al. [4], IoT devices may be susceptible to malware infections, which stealthily propagates between unsecured devices to create massive IoT botnets. The cause of this infection is mainly due to management vulnerabilities. According to CISCO [8], in 2018 about 83% of network devices in their partner sample were running with known vulnerabilities, and IoT devices are implemented without any security planning. These IoT botnets, in turn, are able to execute powerful Distributed Denial of Service (DDoS) attacks. Recently, IoT devices were used on a DDoS attack against the servers of Dyn Inc., a company that controls much of the Internet's DNS infrastructure [9]. This attack is considered to be one of the largest of its kind with a 1.2 Tbps rate.

The traditional DDoS protection approach is the detection and mitigation of the attack at the victim's server or network [10,11], which are able to detect the anomalous traffic pattern and apply mitigation policies. However, as previously discussed, DDoS attacks are becoming more and more powerful, and the attack volume can be larger than the security system is able to handle. A solution for this scenario is proposed by Mirkovic et al. [12], where a system called D-WARD is proposed to deal with DDoS attacks at the source-end network (stub networks or ISP networks). The idea behind this solution is to "divide and conquer", *i.e.*, each ISP network avoids the attack proceeding to the Internet, mitigating the DDoS impact before it reaches its target. As highlighted by the authors, the major challenge of this approach is incentive, since it directly benefits the victims instead of the deploying ISP, for instance.

However, DDoS attacks may also impair the operation of SDN environments since the traffic is managed by a central controller [13,14]. In addition, DDoS attacks may be performed by IoT devices. Finally, SDN is a viable environment to enable the operation of IoT solutions with customized network requirements. We believe that these statements are a great incentive for ISP network to deploy DDoS security systems on source-end networks, targeting the protection of its SDN controller and indirectly mitigating the attack over the Internet.

Thus, we propose a near real-time security system applied on SDN environments to mitigate DDoS attacks originated from inner devices, such as IoT botnets. The proposed system protects the SDN's central controller against flooding and prevents the attack from leaving the source-end network, indirectly protecting the victim's server. The system is divided into two sections, the Detection Module, responsible for detecting and identifying the attack occurrence, and the Mitigation Module, responsible for selecting drop policies to secure the SDN controller.

On the Detection module, we applied a deep learning method using a multidimensional IP flow analysis, called Convolutional Neural Network (CNN) [15]. This method is widely applied on image recognition/classification problems, and provides the system the ability to learn local patterns along the data set.

Although the proposal of a mitigation approach is not in the scope of this paper, we describe how a mitigation approach operates within the presented system.

To measure the efficiency of the presented CNN method on the Detection module, we used two test scenarios. On the first one, we applied SDN data, simulated through the usage of the network emulator Mininet, OpenFlow (IP flow data) and controlled by Floodlight (SDN controller) of different architectures and DDoS intensities. On the second scenario, we tested the CNN on the public DDoS database CicDDoS 2019 [16], since benchmark data sets are a good basis to evaluate and compare the quality of different network anomaly detection methods. Different methods are compared to the CNN on both scenarios for results comparison and data analysis.

The fundamental commitments of this paper are:

- A security system for SDN environments against inward DDoS attacks;
- The system indirectly protects victims' servers by mitigating the DDoS at the source-end network;
- The efficiency evaluation and comparison of distinct fast DDoS detection techniques applied on SDNs.

The remainder of this paper is organized as follows: Section 2 presents a study of related works; Section 3 describes the organization of the proposed system; Section 4 details the CNN method used for anomaly detection on the system's

Detection Module; Section 5 discusses the performance results achieved; Finally, Section 6 presents the conclusions and future works.

2. Related works

Distributed Denial of Service (DDoS) attack is a critical issue in network security that costs organizations and individuals a great deal of time, money, and reputation. Based on this assertion/concern, various techniques for detecting DDoS attacks and reducing its effects in different network environments have been proposed [17]. In [18], the authors used an Artificial Neural Network (ANN) to detect DDoS attacks classifying the traffic into anomalous and genuine. The implemented ANN was trained with old and up-to-date data sets, and it successfully obtained a high detection rate of both known and unknown attacks. Their solution was based on specific feature patterns as the source and destination addresses and ports. Their approach could not handle DDoS attacks where packet headers are encrypted.

Similarly, in [19] the authors detected application-layer DDoS attacks using a Multi-Layered Perceptron (MLP) classification algorithm. The proposed MLP used Genetic Algorithm (GA) as a learning algorithm. A data set generated by a real attack was used during the tests. The findings indicated that important characteristics for attack identification include the number of HTTP GET requests for a particular address in a 20s long time slot, the entropy of the requests, and variance of the entropy. The results showed that the MLP achieved high rates for accuracy and sensitivity.

Wang et al. [20] proposed an approach where raw IP flow data are represented as an image and used Convolutional Neural Network (CNN) to classify and identify malicious traffic. The authors achieved good outcomes on the detection of different malicious events. This representation is a promising approach using CNN. However, by submitting raw flow data on the training process, the method may learn that specific IP addresses are related to malicious behavior.

Liu et al. [21] propose two payload classification approaches based on Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), respectively. These approaches are used for attack detection, and the authors highlight that their ability to learn feature representations without feature engineering from the original data. The authors compare the proposed methods with different approaches, and achieved good results, with accuracy rates higher than 99% in tests using DARPA1998 data set.

Despite such a high detection rate, mitigation of DDoS attack was not discussed in the earlier presented studies. Currently, software-defined networking offers network programmability, making this communication paradigm a trend for traffic controlling [22] and, consequently, contributing to containing this attack. In [23], the authors presented a DDoS-type attack detection and mitigation system suitable for cloud computing environments using SDN structure. Named DaMask, this system is divided into two modules. The first is responsible for detecting the attack, performing statistical analysis. It uses a graphical structure to store the known traffic patterns and, when unusual behavior is observed, this graph determines if malicious connections are occurring. The second module, specifically targeted the dynamic network environment, performing countermeasures, and generating logs about the detected attacks. With a similar purpose, in the mechanism proposed by Cui et al. [24], once a DDoS attack is detected, the anomaly attenuation occurs by inserting entries in the switch table of the first recognized switch in the propagation path of the anomaly. These entries have rules that discard packets whose address and destination port match the attack target.

Joldzic et al. [25] proposed a three layers defense for SDN topology. The outmost is located at the network gateway. This layer has an OpenFlow switch to slice and deliver the resulting chunks of incoming traffic to the subsequent layer. The second layer has several devices called processors, responsible for traffic analysis and anomaly detection. The last layer contains an OpenFlow that aggregates the traffic forwarded by the processors and transmits it to the inside of the network. This approach presents two disadvantages. First, it is assumed that the computer network is free of internal anomalies. In this case, attacks can be launched by hosts inside the network and overwhelm the SDN controller. Second, splitting the traffic into different processors can also lead to the division of the attack between them. This may hinder the search for anomalies since such a division may accidentally mask the attack. Also, to mitigate DDoS attacks, Chen et al. [26] used specialized software boxes to protect the control plane from overloading during the attack by enhancing the ingress switches scalability.

Several works employed push-back schemes to mitigate denial of service attacks [24]. When an attack is detected, the push-back strategy eliminates the attack traffic and notifies other forwarding devices about such traffic. According to Hameed and Ahmed Khan [27], the push-back scheme imposes complexity and overhead in the network management because all forwarding devices in the attack path must be coordinated. To overcome this drawback, they proposed a collaborative DDoS mitigation scheme leveraging SDN. The authors deployed a secure controller-to-controller communication protocol lying in different autonomous systems to transmit attack information with each other. The authors designed a testbed to test three deployment approaches (linear, centralized, and mesh) for policy distribution to other autonomous systems. Experiments showed that mitigation was transferred from destination to source, saving valuable time, and network resources.

In addition to the traditional push-back mitigation scheme, DDoS security solutions were used at the victim-end because of the ease of deployment and availability of complete attack information. Behal et al. [28] argue the lack of sufficient computational resources at the victim-end along with the massive network traffic volume generated by DDoS attacks make security solution itself vulnerable. The authors proposed an ISP level distributed defense system to divide the computational complexity among the nearest point of presence (PoP) routers. Traffic is monitored at all the ingress points of an ISP and sent to a central coordinator in the victim's network.

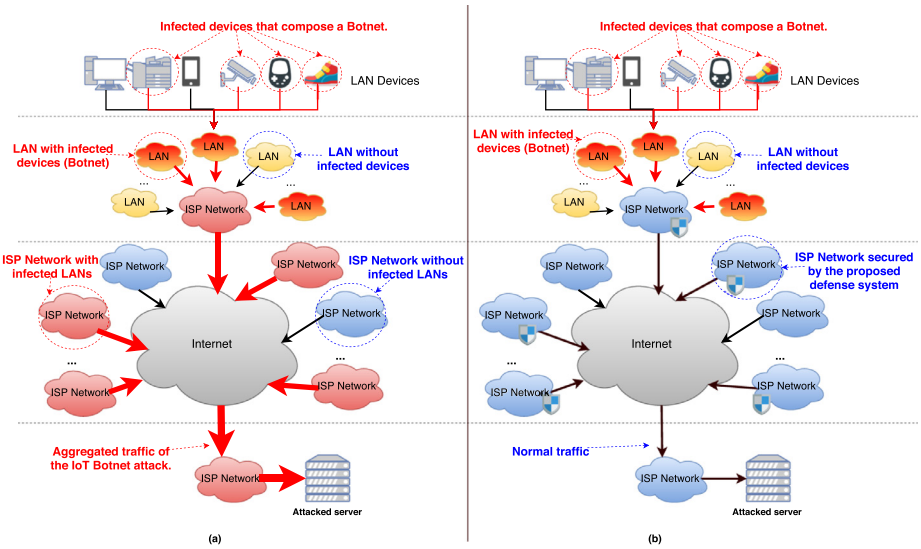

Fig. 1. (a) Representation of a DDoS attack performed by an IoT botnet using different ISP networks. (b) Representation of the previous scenario with DDoS mitigation at the SDN controller of each ISP network.

In this paper, we propose a security system able to protect the SDN controller against internal DDoS attacks targeting an external server. Differently from previous works, our proposal does not require the network infrastructure to be modified. Also, it eliminates the use of push-back to perform mitigation, thereby reducing the complexity of network management. By mitigating the attack at the source-end networks, the overall DDoS attack should be mitigated on the destination-end network, protecting both the targeted server and the SDN controller.

3. Proposed security system

In this section, we depict the functioning of the proposed SDN security system. It is designed to operate within the SDN central controller of ISP networks, helping to protect it against DDoS attacks. Furthermore, by preventing the DDoS attack to proceed to the Internet, the proposed system parallely mitigates DDoS attacks of external targets. This occurs through a "divide and conquer" approach, *i.e.*, the DDoS attack is mitigated inside the source-end network of several different ISP networks. Fig. 1 summarizes this idea.

In Fig. 1(a) represents a DDoS attack without the proposed protection, and (b) represents the attack after the mitigation process. The Internet interconnects several different ISP networks, which, in its turn, connects several LANs composed by heterogeneous devices. With the popularization of IoT solutions, these LANs tend to increase in size and traffic, which consequently make them powerful sources for DDoS attacks since IoT devices may be susceptible to malware infections [4].

These attacks lie upon their distributed architecture to impair the operation of the target's server. As shown in Fig. 1(a), different infected devices within several ISP LANs may target a single server, and the traffic aggregation between this attack and several others from different ISP networks provides a massive resource depletion on the destination-end network. Depending on the amount of infected devices inside the ISP network, this situation may as well impair the operation of its SDN central controller and, consequently, the quality of its provided services to final users.

By preventing these malicious packets from passing through the SDN controller, the DDoS attack is mitigated before it reaches the target's network, as shown in Fig. 1(b). Furthermore, through the usage of dropping policies, malicious traffic will not impair the SDN controller, guaranteeing the ISP network to operate in its normal state.

To provide this protection, the proposed system is based on the analysis of IP flow dimensions, using distinct features to recognize a pattern relating the network's normal operation and to detect the existence of DDoS attacks.

To reduce the DDoS impact over legitimate users, the proposed system operates in near real-time, extracting and analysing IP flow data in one-second intervals. This time interval analysis enables fast detection and mitigation, reducing the damage over both the SDN controller (and consequently its users) and the external attacked server.

It is important to highlight that, in order to enable the speed of the detection and mitigation processes, the system operates autonomously. Thus, even though the system generates an alarm to inform the network administrator when a DDoS is detected, no human interaction is required. A flowchart representing the functioning of the proposed system is described by Fig. 2.

As shown, every second IP flow dimensions or features are exported from the SDN controller through the OpenFlow protocol (6 features in this example). These dimensions are heterogeneous data that can be classified as quantitative (like the rate of packages and bits per second) and qualitative features (like source/destination ports and IP addresses). To enable

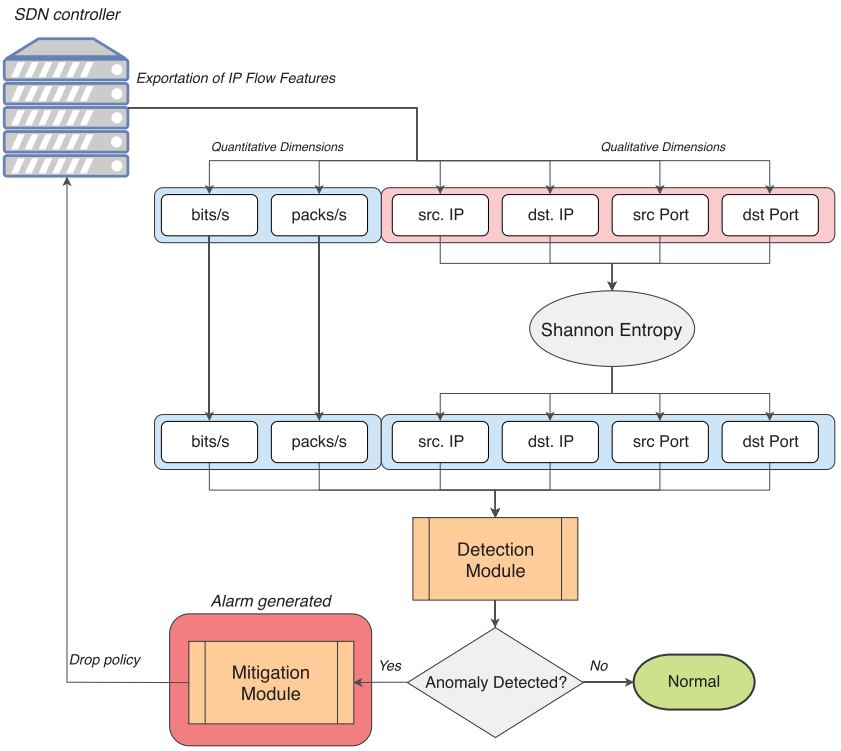

Fig. 2. Overall functioning of the proposed SDN security system.

the usage of this data by the Detection Module, the qualitative dimension must be converted to a quantitative one. Thus, the qualitative dimensions are submitted to the Shannon Entropy, which highlights the concentration or dispersion degree of the analyzed time interval. Given a dimension $X = \{x_1, x_2, ..., xn\}$ in which x_i is the occurrence frequency of the sample i at a given time interval, the Shannon entropy (H) for the dimension X is calculated through:

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

where S is the sum of all the occurrence frequencies of the elements present on the analyzed time interval $\left(S = \sum_{i=1}^{N} x_i\right)$.

Thus, if there is a high concentration of a specific destination IP Address, this dimension will be represented with a low entropy value. On the other hand, if there is a high dispersion in the data relating to the same dimension, a higher entropy value will be retrieved.

After this step, the quantitative data relating to the analysed dimensions are submitted to the Detection Module, responsible for analyzing and detecting the occurrence of DDoS attacks. If a DDoS is detected, then the Mitigation Module is triggered, generating a countermeasure policy that must be incorporated by the SDN controller.

As observed by Fig. 2, the proposed system is composed of modules, and their relation is presented by Fig. 3.

As seen, the proposed security system is separated into two main sections: the Detection and the Mitigation modules, which are composed by complementary sub-modules that contributes to their operation.

As the name suggests, the Detection Module has the objective of detecting and identifying DDoS attacks. In this module, we applied the Convolutional Neural Network approach and compared it with other methods for a performance comparison, which is presented in Section 5. As shown by Fig. 3, the Detection Module is divided into two sub-modules: the “training process” and the “anomaly detection”. As CNN is a supervised learning approach, it requires a prior step of training that will calibrate its classification outcomes. This process will be detailed in Section 4. The Anomaly Detection is the second sub-module, which is in charge of detecting DDoS events on the network over the analyzed IP flow features, as previously described.

The Mitigation Module is triggered when a DDoS event is identified by the Detection Module, being responsible for the decision-making process that will provide the optimal countermeasures for the attack.

As the proposal of a novel mitigation scheme is not on the scope of this paper, we introduce here an approach presented in [29] to illustrate this module’s operation.

Thus, two other sub-modules compose the Mitigation one, the “Game Theoretical countermeasure approach” and the “countermeasure execution”. On the first sub-module, to calculate the optimal countermeasure policy for DDoS mitigation, a

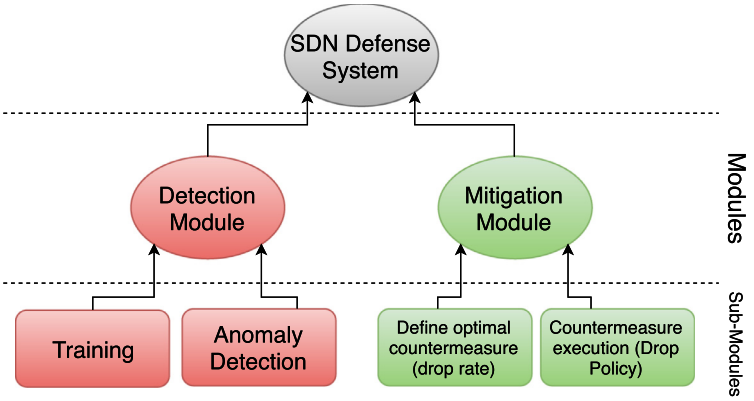

Fig. 3. Modular organization of the SDN security system.

game theory (GT) based approach, described in [29], could be applied, where the process is implemented at an SDN border gateway device and used on the mitigation of DDoS attacks, protecting the network's central controller against internal and external attacks. As previously discussed, these attacks may not be targeting the operation of the SDN itself but, as the communication traffic passes through its central controller, it may be also impaired. The output of this sub-module is an optimal packet drop rate.

The GT approach is a game of two players, the attacking user and the security system. As the DDoS traffic passes through the SDN controller before reaching its real target, for simplicity, we consider the attacker to be targeting the SDN controller. Thus, the attacking user aims to maximize the DDoS impact on the SDN controller while limiting its possibility of being identified. The security system, in turn, tries to limit the impact generated by the attacking user to guarantee the normal operation of the services made available by the SDN (ISP). Such games are classified as "zero-sum games" since the loss of one player is the gain of another, and this gain is commonly defined as payoff. Different metrics are used for the payoff calculation, taking into account i) the error between the analyzed time interval and the expected SDN behavior, ii) the cost of the attack for the attacker player, iii) the consumption of bandwidth by legitimate users compared to attacking ones on the average, iv) and legitimate users' estimated packet drop after the mitigation process [29].

The variables of these metrics are stated through a set of feasible moves executed by the players. The security system is able to:

- Discard packets to avoid their processing;
- Authorize packets to be processed by the central controller.

The attacking user, in turn, is able to:

- Change the attack intensity (amount of packets/s transmitted by each attacking host);
- Change the amount of attacking hosts.

Finally, the second Mitigation sub-module, the "countermeasure execution", provides the SDN controller with the optimal packet dropping policy achieved by the GT approach. In short, the first mitigation sub-module estimates the optimal drop policy, and the second one sends it to the SDN central controller for operation.

4. Anomaly detection approach

We describe in this section the anomaly detection approach applied on the proposed System's Detection Module. A performance comparison between the presented method and others is available on Section 5.

4.1. Convolutional neural network (CNN)

Among the different approaches applied to the detection of computer network attacks and anomalies, deep learning methods are becoming increasingly popular among researchers. Deep learning methods are a subclass of machine learning that is capable of extracting patterns in complex data. Thus, it is widely applied to image recognition and pattern classification problems. As stated by Chollet [15], the "deep" in deep learning stands for the idea of successive layers of representations, as the number of layers representing a method is known as its depth. Deep learning methods commonly use three or more layers of representation, while "shallow" learning methods, like Multi-Layered Perceptron (MLP), focus on learning through only one or two layers.

In [21], the authors highlight that the main benefit of deep learning methods is the absence of manual feature engineering. In other words, the technique is capable of finding patterns among massive data sets during the training process

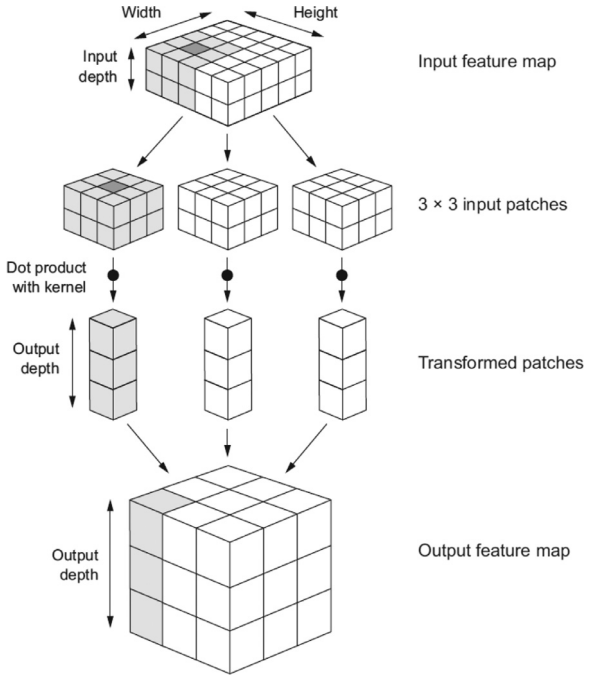

Fig. 4. Operation of the convolution process. [15]

by itself, giving more "importance" to features that are more relevant to the classification process. This characteristic dramatically increases the classification outcomes, since complex patterns, sometimes stealth for human eyes, can be extracted from the data set.

In this paper, we apply a deep learning method known as Convolutional Neural Network (CNN) on the detection of DDoS attacks. As described by Chollet [15], the fundamental difference between a fully connected layer (used by MLPs, for instance) and a convolutional layer is that the first learns global patterns in their input feature space, while the second is capable of learning local patterns. As CNN's are commonly applied to image processing environments, it can extract local patterns on the image, which significantly improves the accuracy of classification problems.

This precision is possible through the convolutional operations that compose CNN. A convolution is an operation between two functions that produces a third one, which expresses how the shape of one is modified by the other. As described by Chollet [15], convolutions operate over 3D tensors called *feature maps*. On image classification problems, for instance, they stand for two spatial axes (width and height) and a *channel* axis (for RGB images, the channel is 3, wherein black-and-white images, it is 1). To convolve this input, a *filter* is applied through dot products to extract local patterns. The filter operates like a sliding window, performing a dot product with all the unique positions where it can be put on the image, encoding specific characteristics of the input. In other words, a convolution works by sliding these filters of fixed size over the 3D input feature map, stopping at every possible location, and extracting the 3D patches, which are processed via dot products into 1D dimension outputs. These outputs, in turn, are reassembled into a 3D output map, as described in Fig. 4.

On CNN networks, Convolutional layers are commonly followed by Pooling layers, whose objectives are to reduce the spatial size of the representation. Thus, the pooling process reduces the number of parameters and computation in the CNN, *i.e.*, downsample feature maps. The most common approach used in this layer is the Max pooling, which consists of extracting windows from the input feature maps and outputting the max value of each channel [15], due to its efficiency.

However, IP flow traffic data are represented as a time series, not an image. Thus, a variation of the traditional 2D convolutional operation is used in this paper, the 1D-CNN [15]. In a straightforward comparison, it operates with the same structures and functions, but through 1-dimensional data (time series), like show by Fig. 5.

1D-Convolutional layers also receive 3D tensors as input: the first one representing the number of samples, the second standing for the time, and the third for the features [15]. In this paper, as the system is operating with one-second data, the input tensor is configured as (samples, features, channels). The architecture of the CNN implemented in this paper is described by Fig. 6.

As observed, the architecture of the CNN is composed of a stack of two *Conv1D* and *MaxPooling1D* layers. They are followed by a *Flatten* layer, responsible for transforming the 3D output of the previous layers into 2D inputs for the following layers, a *Dropout* layer, aiming to avoid over-fitting as CNN's tend to converge very fast to a solution, and a *Dense* or *Fully-Connected* layer, to perform a global model classification. The output is a single neuron with a sigmoid activation function for binary classification, *i.e.*, which classifies data as normal or DDoS.

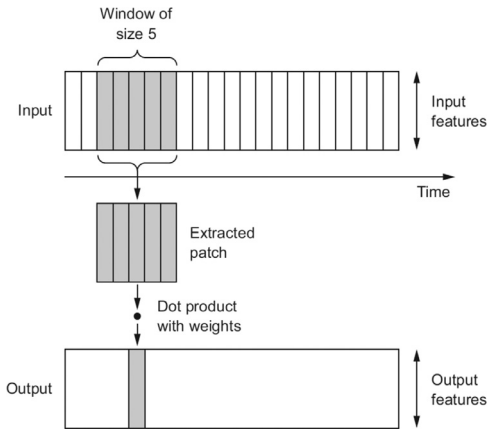

Fig. 5. Convolution in one dimension. [15]

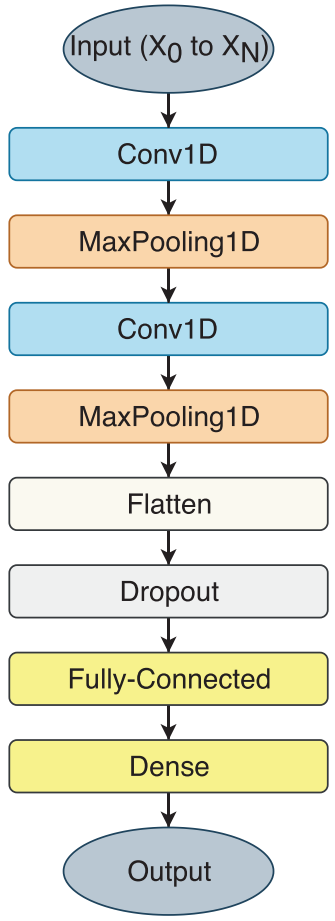

Fig. 6. CNN architecture.

5. Performance outcomes

We examine in this section the performance results relating to the proposed detection module for the security system. For this, we tested the system over two different scenarios and compared the CNN approach with different methods for performance evaluation. These methods are: the Multi-Layered Perceptron (MLP) Network, a machine learning method with one hidden layer composed of 10 neurons; the Deep Neural Network (DNN) or Dense MLP (D-MLP) [15,30], the deep learning version of the MLP, containing 3 hidden layers with 10 neurons on each one; and the Logistic Regression [13], which is a

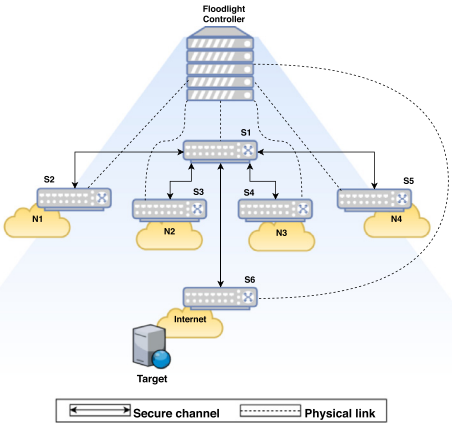

Fig. 7. Simulated SDN topology.

Table 1
Summary of training (Day 1) and test days (Days 2 to 7) on the first scenario.

Day 4	Day 5	Day 6	Day 7				
Switches	6	6	6	6	6	6	6
Hosts	120	200	200	150	150	150	150
# of DDoS attacks	2	1	1	1	1	1	1 (short)
# of attacking hosts	15	20	20	15	20	10	10

statistical model used to predict values taken by a categorical variable from a series of continuous or binary explanatory variables. All detection methods were implemented using Python and Keras on a computer using Windows 10 64bit, Intel Core i7 2.8GHz, and 8GB of RAM.

5.1. Scenario 1 - SDN simulated data

In this scenario, we applied simulated IP flows, generated through the network emulator Mininet, to perform the tests. Mininet is a lightweight software that enables the generation of realistic emulated SDN environments composed by hosts, links, switches, and controllers. One of its main advantages is the ease of the production of customized network topologies through a virtual machine. To control the simulated network switches, we used the Open vSwitch, which is compatible with the Mininet environment. Together with Mininet, we applied the Floodlight, an SDN controller extensively utilized in literature, where the proposed security system is implemented. The OpenFlow protocol performs data collection.

Six switches compose the SDN environment used for the system's analysis. As one central device interconnects the others, all the five remaining switches control from 24 to 40 hosts, totaling 120 to 200 connected hosts, respectively. One of these switches stands for a boundary gateway device, containing the external (Internet) hosts and the attacked server. Fig. 7 describes the network topology.

Seven days of SDN traffic (168 hours) were generated and used for both training and testing. Each day interval emulates the normal behavior of an ISP network, with higher data traffic on working hours and by the evening, while lower data traffic occurs early in the morning and by dawn.

Two distinct-intensity occurrences of DDoS attacks were injected on the data of the first generated day (Day 1), which was utilized for training and adjustment of the methods' anomaly detection procedures.

The following six generated days (Days 2 to 7) were applied to test the performance of the presented DDoS detection methods. A summary of each one of the training and test days is shown in Table 1. As observed, the testing days have different characteristics in comparison to the training day, all of them with a higher number of hosts, which should make the attacks stealthier, *i.e.*, harder to detect. Days 2 and 3 have an increased number of attacking hosts on different periods of the day. On days 3 and 4, we applied 150 hosts on the network, where 15 and 20 of them are malicious ones, respectively. Finally, Days 6 and 7 use 150 hosts on the architecture, in which 10 of them are malicious nodes. These are stealthier scenarios, in which their only difference is the attack duration: while the attack on Day 6 lasts for 3638 time intervals, the same attack lasts for 618 time intervals on Day 7.

In this first scenario, we exported six IP flow dimensions from the SDN controller, which are: Bits and Packets per second, Source and Destination IP addresses and Ports. With this amount of analyzed features, the number of computed parameters is low. So, we suppressed the first MaxPooling layer of Fig. 6 to avoid data loss. The parameters were set with 16 and 8 filters for the first and second Conv1D layers, respectively, both with kernel (filter) size 3. The second MaxPooling1D was

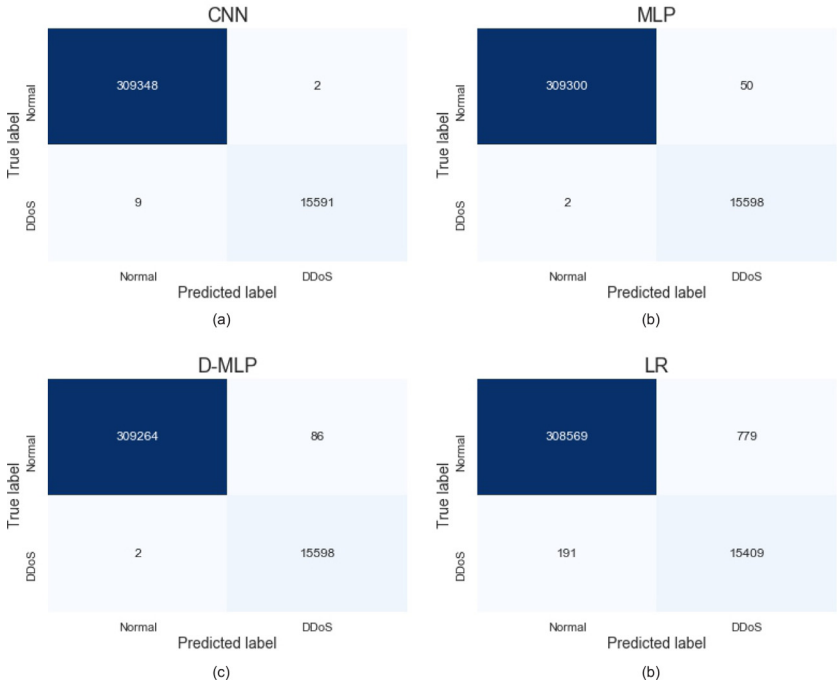

Fig. 8. Confusion Matrices of the tested methods relating the first test scenario.

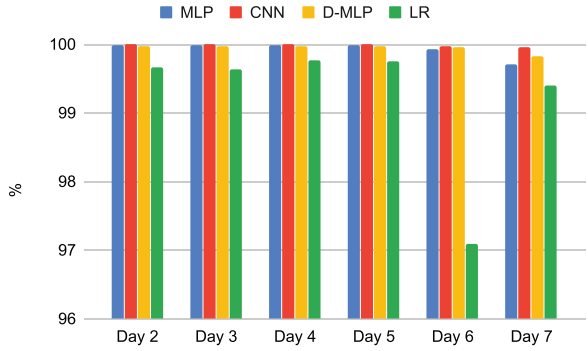

Fig. 9. Methods' accuracy outcomes for the first test scenario.

set with a pool size of 2, and the Dropout with a rate of 0.5. The Fully-connected layer is composed of 10 neurons, and the output layer is formed of 1 neuron to generate a binary result. All methods were tested through 1000 epochs.

We applied classical anomaly detection statistic techniques to measure their efficiency in detecting DDoS attacks targeting an external server. Fig. 8 shows the outcomes achieved by each one of the tested methods through confusion matrices.

As observed, the CNN method fared better, achieving the lowest false-positive (FP) rate, *i.e.*, when benign data are classified as a DDoS attack, followed by MLP, D-MLP, and LR methods. MLP and D-MLP methods achieved lowest false-negative (FN) rates, *i.e.*, when a DDoS interval is classified as normal, but the difference between them and CNN is small.

Other classical metrics used to measure anomaly detection performance are the accuracy, precision, recall and f-measure techniques. Accuracy shows the percentage of time intervals correctly classified. Precision estimates the ratio of intervals correctly recognized as DDoS among all the samples classified as DDoS. The recall metric represents the percentage of correctness for DDoS intervals. Finally, F-measure represents the harmonic mean between recall and precision. The results achieved by the tested methods using these metrics on each day are shown by Figs. 9–12.

In this test scenario, despite the results presented by the confusion matrices, a global analysis of the accuracy rates, presented by Fig. 9, shows that all methods achieved good classification results on the average. The accuracy rates were higher than 99.5% for all tested approaches for all analyzed days, except for the LR method on Day 6. CNN method presented better results in comparison to the other methods, despite the similarity of their outcomes (rates around 99.9% on the average), as the difference between CNN, MLP, and D-MLP accuracies are around 0.01%. The LR method achieved the lowest accuracy results, with rates around 0.03% lower than CNN on the average. This method achieved an accuracy rate of around

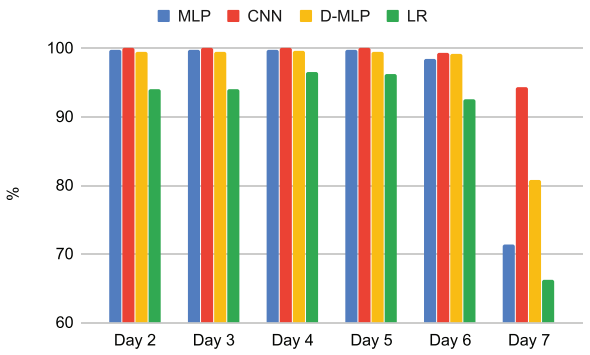

Fig. 10. Methods' precision outcomes for the first test scenario.

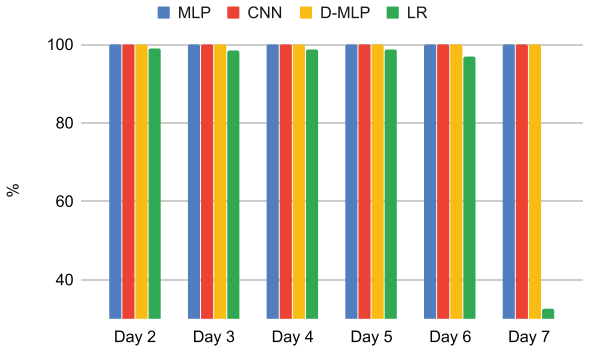

Fig. 11. Methods' recall outcomes for the first test scenario.

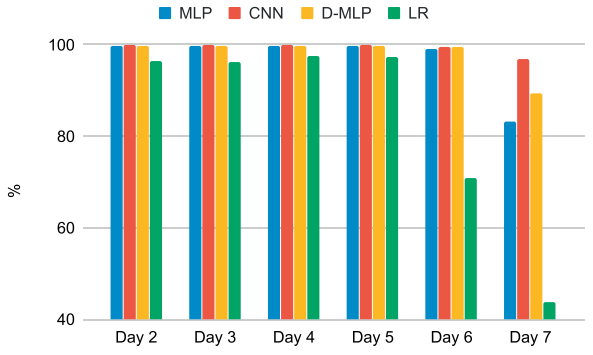

Fig. 12. Methods' f-measure outcomes for the first test scenario.

97% on Day 6, a day with a stealthier attack. However, on Day 7, the LR method achieved an improved accuracy outcome, even though both days present the same attack intensity. This occurrence is due to the attack duration since Day 7 has a shorter malicious time interval than Day 6.

As shown by Fig. 10, the CNN method achieved better outcomes for the precision metric, with a rate of 99.9% on the average. This result was expected, as the analysis of the confusion matrices pointed out that this method is more efficient in classifying benign intervals. Furthermore, MLP achieved precision rates of 99.7% on the average, faring slightly better than its deep-learning approach, which achieved precision rates of 99.4% on the average, possibly due to an overfitting occurrence. As pointed out by Chollet [15], when a small amount of data is available, models with fewer layers tend to be more efficient. Finally, the LR method fared worse, achieving better results on Days 4 and 5, when the number of hosts and attack intensity are nearer to the training set, with a precision rate of 95.1% on the average. However, most methods achieved low precision rates on Day 7, which is due to the day's characteristics (low DDoS intensity for a short period). On this day, CNN achieved a precision rate of 94.2%, which is lower than the results regarding the other analyzed days but significantly higher than the rates achieved by the different methods. Thus, it is possible to infer that the CNN method efficiently extracted the main characteristics of both benign and malicious traffic.

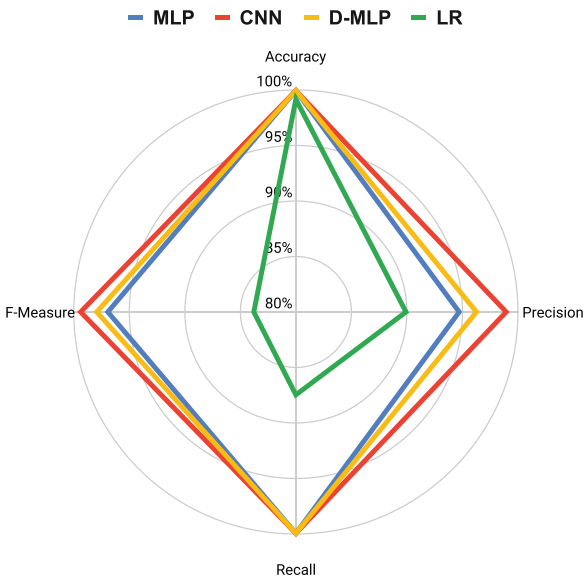

Fig. 13. Radar plot presenting the methods' average outcomes for the first test scenario.

As previously cited, the recall metric shows the efficiency in identifying DDoS intervals. The results presented by Fig. 11 show that CNN, MLP, and D-MLP achieved very similar results, with rates of around 99.9%, even though CNN fared slightly worse (with recall rates 0.04% lesser than MLP and D-MLP methods). Although LR fared worse than the other approaches, it achieved recall values higher than 96.7% on days 2 to 6, which is a good outcome. However, this method produced a low recall rate on Day 7, which points out that it could not efficiently identify the attack with low intensity performed over a short period in this scenario.

Finally, Fig. 12 presents the methods' outcomes relating to the f-measure metric. As it represents the harmonic mean between precision and recall metrics, CNN fared slightly better than the other techniques on Days 2 to 6, with rates of 99.9% on the average, followed by MLP, D-MLP, and LR methods, which achieved average rates of 99.7%, 99.6% and 91.7%, respectively. However, on Day 7, due to the test day's characteristics, the f-measure achieved by the tested approaches differs the most, with rates of 97%, 89.3%, 83.3%, and 43.8% for the methods CNN, D-MLP, MLP, and LR, respectively. Although all the tested approaches were able to detect DDoS attacks on the analyzed days efficiently, only the CNN method consistently performs this task on all evaluated days, which proves its efficiency in comparison to the other tested approaches.

Fig. 13 presents a radar plot that summarizes the results previously addressed in a single image, where the nearer to the outer circle, the closer to 100% the analyzed approach fared on the average for the four measurement techniques. While CNN, MLP, and D-MLP reached similar outcomes for accuracy and recall, CNN achieved better results for precision and f-measure rates.

Thus, in this test scenario, it is possible to conclude that the CNN method achieved the best classification results, operating as an efficient DDoS identifier at the Detection Module of the presented SDN defense system.

5.2. Scenario 2 - CICDDoS 2019 Data set

In this scenario, we applied simulated IP flows collected from a public data set called CICDDoS 2019 [16]. It generates realistic background traffic profiled through B-Profile System to abstract the behavior of human interactions for benign traffic. In this data set, the authors abstract the behavior of 25 users based on different protocols, such as HTTP, FTP, and SSH.

The CICDDoS 2019 data set separates the data into two days. The first one is a training day, containing 12 types of different DDoS attacks, including NTP, DNS, MSSQL, LDAP, NetBIOS, SNMP, UDP, UDP-Lag, SSDP, Syn, WebDDoS and TFTP. The second is a testing day, containing 6 different DDoS attacks, which are NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, and Syn.

This data set provides data with 87 extracted IP Flow features, such as source and destination IP addresses and ports, protocols, several flags, counters, and flow identification features. All the data was submitted to a data formatting process to convert qualitative features into quantitative ones, and to group data into one-second intervals like described in Section 3.

The parameters of the CNN were set with 64 and 32 filters and a kernel size of 32 and 16 for the first and second Conv1D layers, respectively. Both MaxPooling1D layers were set with a pool size of 2, and the Dropout with a rate of 0.5. The Fully-connected layer is composed of 10 neurons, and the output layer is formed of 1 neuron to generate a binary result. All methods were tested through 1000 epochs.

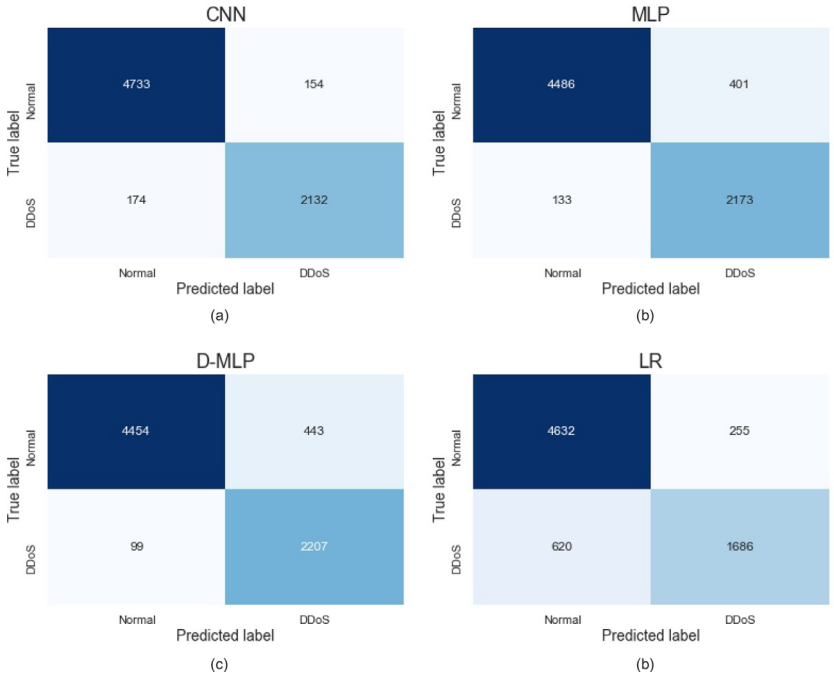

Fig. 14. Confusion Matrices of the tested methods relating the first test scenario.

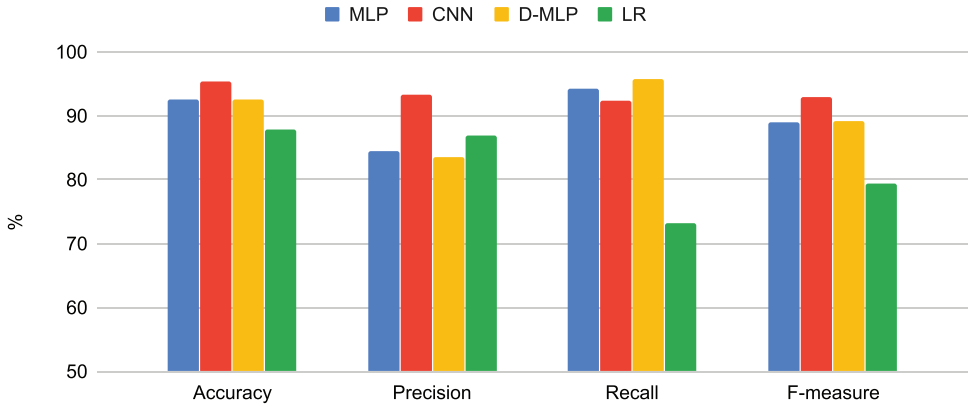

Fig. 15. Methods' outcomes for accuracy, precision, recall and f-measure metrics over the second test scenario.

The efficiency measurement was performed as described in the first test scenario, comparing CNN with the methods MLP, D-MLP, and LR over classical techniques. Fig. 14 shows the outcomes achieved by each one of the tested methods through confusion matrices on this scenario.

The second test scenario presents some interesting differences from the first one, which can be observed at the Confusion Matrices presented by Fig. 14. Besides having less normal intervals, the DDoS attacks present in this data are related to 12 different behaviors, which makes the binary classification process a complex task. As observed by the confusion matrices, the number of false-positive and false-negative intervals is higher for all the methods in comparison to the ones achieved in the first test scenario. With relation to the number of false-positive intervals, the CNN method fared better, followed by LR, MLP, and D-MLP, respectively. When considering the false-negative intervals, the D-MLP fared better, followed by MLP, CNN, and LR methods. While MLP and D-MLP presented a greater difficulty in classifying normal intervals, the LR method faired worse on classifying DDoS intervals. The CNN method, in turn, presented the most balanced outcome.

The results achieved by the tested methods using classical metrics in this scenario are shown in Fig. 15

As observed in Fig. 15, the CNN method obtained better accuracy measures, reaching a rate of 95.4%, followed by D-MLP, MLP, and LR approaches, which achieved rates of 92.6%, 92.5% and 87.8%, respectively.

For the precision metric, the CNN approach also fared better, with a rate of 93.3%, followed by LR, MLP, and D-MLP methods, with 86.8%, 84.4% and 83.4% precision rates, respectively.

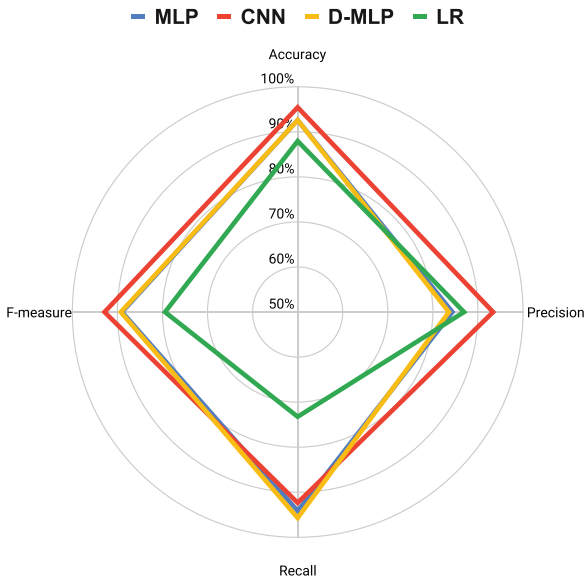

Fig. 16. Radar plot presenting the methods' average outcomes for the second test scenario.

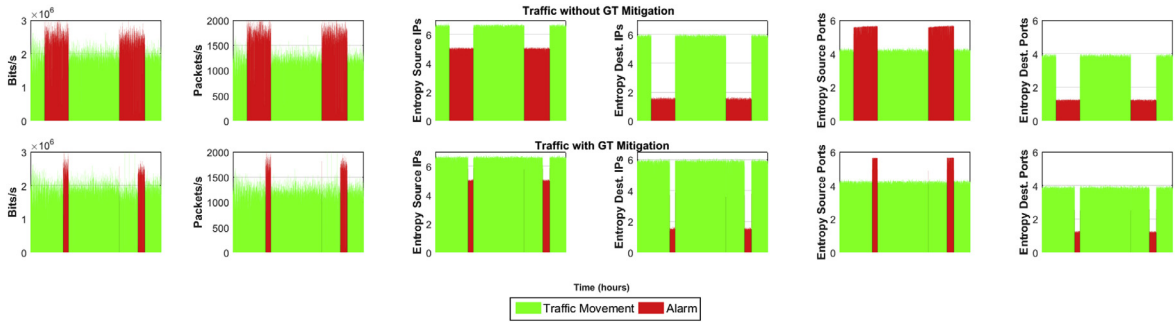

Fig. 17. Hexa-dimensional IP flow view of the analyzed SDN, with two DDoS attacks based on 15 hosts, before and afterwards the procedures of detection and mitigation utilizing CNN and GT.

For the recall metric, the D-MLP achieved the best results, with a rate of 95.7%, followed by MLP, with a percentage of 94.2%, CNN, which reached a 92.4% recall measure, and LR, with a rate of 77.1%.

Finally, for the f-measure metric, the CNN method achieved better outcomes with a rate of 92.8%, followed by D-MLP, MLP, and LR methods, which obtained an f-measure result of 89.2%, 89% and 79.4%, respectively.

Fig. 16 summarizes the previously addressed metrics' results in a single figure. In short, CNN methods achieved better accuracy, precision, and f-measure results than the other tested methods, reaching values around 95%. In turn, D-MLP presented the best recall outcome, followed by the MLP method, which produced similar results using fewer layers.

As the results achieved in the first scenario, the CNN method also fared better on the average than the other approaches on the second test environment, achieving promising test outcomes that make it an efficient technique on DDoS detecting.

5.3. Mitigation

As previously discussed in Section 3, the proposal of a novel mitigation approach is not in the scope of this paper. So, to demonstrate the efficiency of the presented SDN security system, we applied a game-theoretical (GT) approach, proposed in [29], on the mitigation module. This method is used in DDoS mitigation of internal attacks against external targets.

The Mitigation module is in charge of providing the SDN central controller with the optimal drop policy, aiming to mitigate or even interrupt the DDoS attack entirely. By preventing the distributed attack from reaching the Internet, it is possible to mitigate the attack on the destination-end network indirectly.

Every second, IP flow data are collected and submitted to the Detection module, where a classification process occurs based on the anomaly detection method. This classification outcome may be a "normal" label or a "DDoS" one, in which the Mitigation module is triggered. Thus, the GT-approach analyses the provided information to automatically determine the optimal drop rate as a DDoS countermeasure. Fig. 17 shows the traffic of the analyzed SDN before and after the mitigation

process, relating the first test day of the first scenario through the usage of the CNN anomaly detection method. For better understanding, the figure shows the traffic from 12:30 to 19:30, the interval in which the DDoS attacks occurred.

As observed, the GT-approach retrieved the optimal drop policy, which was incorporated by the SDN controller at the next time interval, *i.e.*, at the upcoming second from the CNN detection. It is also possible to observe that the final stage of the attacks was not mitigated. It occurs because the drop policy generated to minimize the DDoS has a lifespan of one hour, and the attacks lasted longer. When this happens, a new alarm will be generated by the detection module, and the GT-approach will be triggered once again, restarting the process.

As the results point out, the GT-approach was able to mitigate the DDoS attacks successfully. The mitigation module brings the SDN back to its regular operation, and GT represents a feasible approach against both internal and external DDoS attacks.

6. Conclusions

The proposed defense system was capable of inspecting the SDN traffic behavior in one-second time intervals. It effectively detects and mitigates the occurrence of DDoS attacks on the controller and, consequently, over the external targeted server.

We presented a Convolutional Neural Network (CNN) approach to acting within the Detection module, which was tested against three other anomaly detection approaches: the Logistic Regression (LR); the Multi-Layered Perceptron (MLP) network; and Dense MLP. The methods were submitted to two test scenarios. The first one uses simulated SDN data, generated using Mininet and Floodlight, over four different days containing DDoS attacks. The second one uses a public data set known as CicDDoS 2019, providing more than 12 types of DDoS attacks. As the LR method fared worse on most tests, the other three tested methods achieved relatively good results. The CNN method achieved low false-positive rates, higher accuracy, precision, and f-measure outcomes. The MLP and D-MLP methods fared better for the recall metric, achieving similar results.

Moreover, we described the Mitigation module and presented an example of an operation using a game-theoretical approach. To mitigate the attack, we applied a Game Theory (GT) based technique that optimizes the packet discard rate in a policy applied inside the central controller of the SDN. The outcomes reveal that the mitigation approach is efficient in restoring the SDN's regular operation.

For future works, we intend to increase the number of hosts on the simulated SDN environment to test the behavior of the proposed system against stealthier DDoS internal attacks. Furthermore, we want to study the impact of recurrent deep learning approaches, such as LSTM and GRU, on classification problems such as DDoS detection in SDN environments.

Declaration of Competing Interest

Authors declare that they do not have any conflicto of interest

CRediT authorship contribution statement

Marcos V.O. de Assis: Conceptualization, Data curation, Formal analysis, Writing - original draft. **Luiz F. Carvalho:** Conceptualization, Data curation, Formal analysis, Writing - original draft. **Joel J.P.C. Rodrigues:** Conceptualization, Data curation, Formal analysis, Writing - original draft. **Jaime Lloret:** Conceptualization, Data curation, Formal analysis, Writing - original draft. **Mario L. Proença Jr:** Conceptualization, Data curation, Formal analysis, Writing - original draft.

Acknowledgments

This study was financed in part by the National Council for Scientific and Technological Development (CNPq) of Brazil under Grants 310668/2019-0 and 309335/2017-5; by the Ministerio de Economía y Competitividad in the "Programa Estatal de Fomento de la Investigación Científica y Técnica de Excelencia, Subprograma Estatal de Generación de Conocimiento" within the project under Grant TIN2017-84802-C2-1-P; by FCT/MCTES through national funds and when applicable co-funded EU funds under the Project UIDB/EEA/50008/2020; and by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) by the granting of a scholarship through the "Programa de Doutorado Sanduche no Exterior (PDSE) 2019". Finally, this work was supported by Federal University of Paraná (UFPR) under Project Banpesq/2014016797.

References

[1] Tudosa I, Picariello F, Balestrieri E, De Vito L, Lamonaca F. Hardware security in IoT era: the role of measurements and instrumentation. In: 2019 II workshop on metrology for industry 4.0 and IoT (MetroInd4.0 IoT); 2019. p. 285–90. doi:10.1109/METROI4.2019.8792895.

[2] El-Mougy A, Ibnkahla M, Hegazy L. Software-defined wireless network architectures for the internet-of-things. In: 2015 IEEE 40th local computer networks conference workshops (LCN Workshops); 2015. p. 804–11. doi:10.1109/LCNW.2015.7365931.

[3] Chaabouni N, Mosbah M, Zemmari A, Sauvignac C, Faruki P. Network intrusion detection for IoT security based on learning techniques. IEEE Commun Surv Tut 2019;21(3):2671–701. doi:10.1109/COMST.2019.2896380.

[4] Kim H, Kim T, Jang D. An intelligent improvement of internet-wide scan engine for fast discovery of vulnerable IoT devices. Symmetry 2018;10(5). doi:10.3390/sym10050151.

[5] Bizanis N, Kuipers FA. Sdn and virtualization solutions for the internet of things: a survey. IEEE Access 2016;4:5591–606. doi:10.1109/ACCESS.2016.2607786.

[6] Caraguay A, Peral A, López L, Villalba L. SDN: evolution and opportunities in the development IoT applications. Int J Distrib SensNetw 2014;10(5):735142. doi:10.1155/2014/735142.
[7] Rego A, Garcia L, Sendra S, Lloret J. Software Defined Network-based control system for an efficient traffic management for emergency situations in smart cities. Fut Gener Comput Syst 2018;88:243–53. doi:10.1016/j.future.2018.05.054.
[8] CISCO. Annual cybersecurity report. 2018. Accessed 2 February 2019; https://www.cisco.com/c/en/us/products/security/security-reports.html.
[9] Kolias C, Kambourakis G, Stavrou A, Voas J. Ddos in the IoT: Mirai and other botnets. Computer 2017;50(7):80–4. doi:10.1109/MC.2017.201.
[10] Proenca ML, Zarpelao BB, Mendes LS. Anomaly detection for network servers using digital signature of network segment. In: Advanced industrial conference on telecommunications/service assurance with partial and intermittent resources conference/e-learning on telecommunications workshop (AICT/SAPIR/ELETE'05); 2005. p. 290–5. doi:10.1109/AICT.2005.26.
[11] Semerci M, Cemgil AT, Sankur B. An intelligent cyber security system against DDoSattacks in sip networks. Comput Netw 2018;136:137–54. doi:10.1016/j.comnet.2018.02.025.
[12] Mirkovic J, Prier G, Reiher P. Attacking DDoS at the source. In: 10th IEEE International conference on network protocols, 2002. Proceedings.; 2002. p. 312–21. doi:10.1109/ICNP.2002.1181418.
[13] Carvalho LF, Abrão T, de Souza Mendes L, Proença ML. An ecosystem for anomaly detection and mitigation in software-defined networking. Expert Syst Appl 2018;104:121–33. doi:10.1016/j.eswa.2018.03.027.
[14] Mladenov B. Studying the DDoS attack effect over SDN controller southbound channel. In: 2019 X National conference with international participation (ELECTRONICA); 2019. p. 1–4. doi:10.1109/ELECTRONICA.2019.8825601.
[15] Chollet F. Deep learning with python. 1st . Greenwich, CT, USA: Manning Publications Co.; 2017. ISBN 1617294438, 9781617294433.
[16] Sharafaldin I, Lashkari A, Hakak S, Ghorbani A. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In: 2019 IEEE 53rd International Carnahan Conference on Security Technology; 2019.
[17] Fernandes Jr G, Rodrigues JJ, Carvalho LF, Al-Muhtadi JF, Proença Jr ML. A comprehensive survey on network anomaly detection. Telecommun Syst 2019;70(3):447–89. doi:10.1007/s11235-018-0475-8.
[18] Wang Z. An elastic and resiliency defense against DDoS attacks on the critical DNS authoritative infrastructure. J Comput Syst Sci 2019;99:1–26. doi:10.1016/j.jcss.2017.05.012.
[19] Johnson Singh K, Thongam K, De T. Entropy-based application layer DDoS attack detection using artificial neural networks. Entropy 2016;18(10). doi:10.3390/e18100350.
[20] Wang Y, An J, Huang W. Using CNN-based representation learning method for malicious traffic identification. In: 2018 IEEE/ACIS 17th international conference on computer and information science (ICIS); 2018. p. 400–4. doi:10.1109/ICIS.2018.8466404.
[21] Liu H, Lang B, Liu M, Yan H. CNN and RNN based payload classification methods for attack detection. Knowl-Based Syst 2019;163:332–41. doi:10.1016/j.knosys.2018.08.036.
[22] Jimenez JM, Romero O, Lloret J, Diaz JR. Energy savings consumption on public wireless networks by SDN management. Mob Netw Appl 2016. doi:10.1007/s11036-016-0784-7.
[23] Wang B, Zheng Y, Lou W, Hou YT. DDoS attack protection in the era of cloud computing and Software-Defined Networking. Comput Netw 2015;81:308–19. doi:10.1016/j.comnet.2015.02.026.
[24] Cui Y, Yan L, Li S, Xing H, Pan W, Zhu J, et al. SD-Anti-DDoS: fast and efficient DDoS defense in software-defined networks. J Netw Comput Appl 2016;68:65–79. doi:10.1016/j.jnca.2016.04.005.
[25] Joldzic O, Djuric Z, Vuletic P. A transparent and scalable anomaly-based DoS detection method. Comput Netw 2016;104:27–42. doi:10.1016/j.comnet.2016.05.004.
[26] Chen K, Junuthula AR, Siddhrau IK, Xu Y, Chao HJ. SDNShield: towards more comprehensive defense against DDoS attacks on SDN control plane. In: 2016 IEEE conference on communications and network security (CNS); 2016. p. 28–36. doi:10.1109/CNS.2016.7860467.
[27] Hameed S, Ahmed Khan H. SDN based collaborative scheme for mitigation of DDoS attacks. FutInternet 2018;10(3). doi:10.3390/fi10030023.
[28] Behal S, Kumar K, Sachdeva M. D-FACE: an anomaly based distributed approach for early detection of DDoS attacks and flash events. J Netw Comput Appl 2018;111:49–63. doi:10.1016/j.jnca.2018.03.024.
[29] Assis MVOD, Hamamoto AH, Abrão T, Proença ML. A game theoretical based system using holt-winters and genetic algorithm with fuzzy logic for DoS/DDoS mitigation on SDN networks. IEEE Access 2017;5:9485–96. doi:10.1109/ACCESS.2017.2702341.
[30] Abdulhammed R, Faezipour M, Abuzneid A, AbuMallouh A. Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic. IEEE Sens Lett 2019;3(1):1–4. doi:10.1109/LSENS.2018.2879990.

Marcos Vinicius Oliveira de Assis is a professor of the Federal University of Paraná, Brazil. He received a M.Sc. in Computer Science at the State University of Londrina Brazil and is a Ph.D. student in Electrical Engineering at the same institution. He worked as a visiting researcher at the Polytechnic University of Valencia Spain. He is part of the research group "Computer Networks and Data Communication."

Luiz Fernando Carvalho received the Ph.D. degree in Electrical Engineering and Telecommunications from State University of Campinas in 2018. He completed his masters degree in Computer Science at State University of Londrina in 2014. 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 are management and security of computer networks.

Joel J. P. C. Rodrigues is a professor at the Federal University of Piauí, Brazil; senior researcher at the *Instituto de Telecomunicações*, Portugal; and collaborator of the Post-Graduation Program on Teleinformatics Engineering at the Federal University of Ceará (UFC), Brazil. He is the leader of the Next Generation Networks and Applications (NetGNA) research group (CNPq), an IEEE Distinguished Lecturer, and Fellow of IEEE.

Jaime Lloret received his B.Sc.+M.Sc. in Physics in 1997, his B.Sc.+M.Sc. in electronic Engineering in 2003 and his Ph.D. in telecommunication engineering (Dr. Ing.) in 2006. He is currently Associate Professor in the Polytechnic University of Valencia. Since 2016 he is the Spanish researcher with highest h-index in the TELECOMMUNICATIONS journal list according to Clarivate Analytics Ranking. He is an IEEE Senior, ACM Senior and IARIA Fellow.

Mario Lemes Proença Jr. is an Associate Professor and leader of the research group that studies computer networks in the Computer Science Department at State University of Londrina (UEL), Brazil. He received the Ph.D. degree in Electrical Engineering and Telecommunications from State University of Campinas (UNICAMP) in 2005. He received a M.Sc degree in Computer Science from the Informatics Institute of Federal University of Rio Grande do Sul (UFRGS), in 1998.