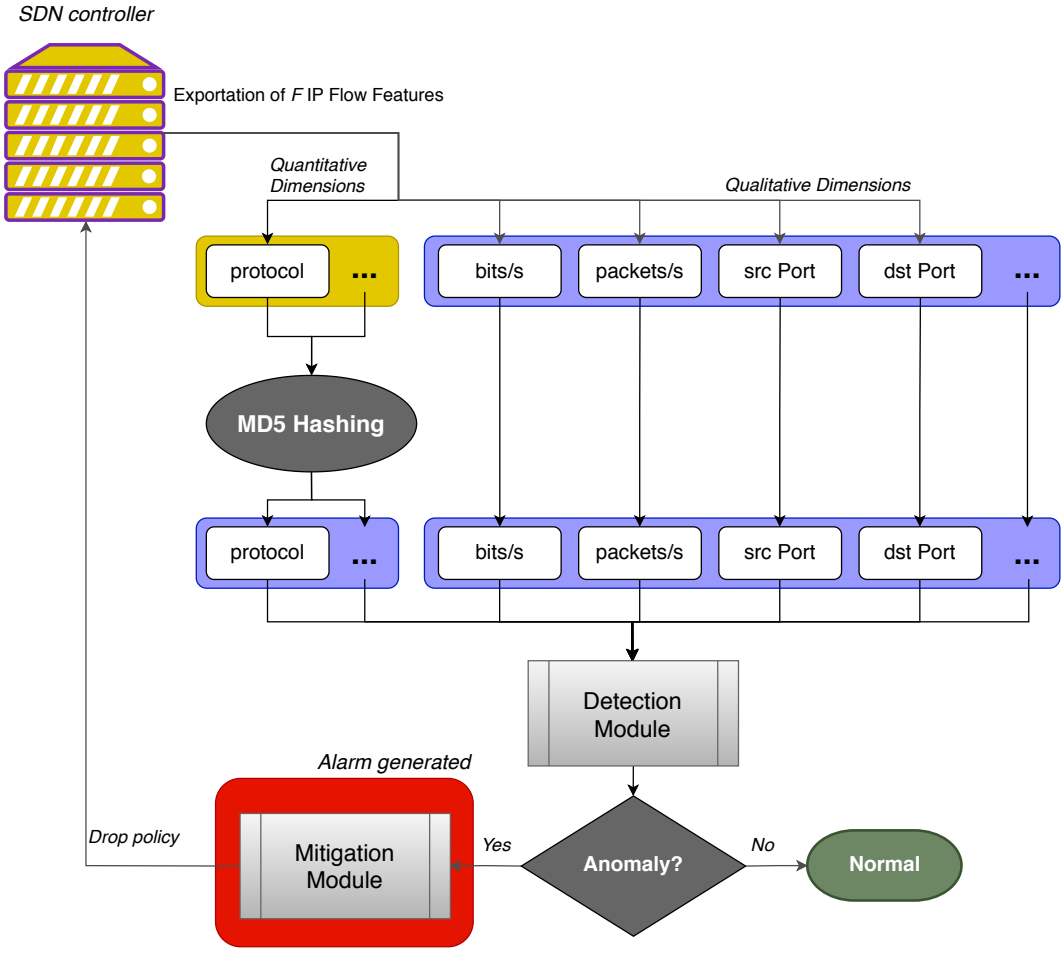
Graphical Abstract

A GRU Deep Learning System against Attacks in Software Defined Networks

Marcos V. O. Assis, Luiz F. Carvalho, Jaime Lloret, Mario L. Proença Jr.



Overall operation of the proposed SDN security system, which aims to protect its central controller against intrusion and DDoS attacks through individual IP flow analysis.

Highlights

A GRU Deep Learning System against Attacks in Software Defined Networks

Marcos V. O. Assis, Luiz F. Carvalho, Jaime Lloret, Mario L. Proença Jr.

- This paper introduces a system for SDN's defense against intrusion and DDoS attacks;
- We propose an anomaly detection scheme based on isolated flow analysis using GRU;
- We present an efficiency evaluation of distinct detection techniques applied to SDNs;
- We used public datasets for performance analysis, which enable results' replication.

A GRU Deep Learning System against Attacks in Software Defined Networks

Marcos V. O. Assis[a], Luiz F. Carvalho[b], Jaime Lloret[d], Mario L. Proença Jr.[c]

[a]*Department of Engineering and Exacts, Federal University of Paraná, Brazil*
[b]*Federal University of Technology - Paraná, Brazil*
[c]*Computer Science Department, State University of Londrina, Paraná, Brazil*
[d]*Integrated Management Coastal Research Institute, Universitat Politecnica de Valencia, Valencia, Spain*

Abstract

The management of modern network environments is becoming more and more complex due to new requirements of devices' heterogeneity regarding the popularization of the Internet of Things (IoT), as well as the dynamic traffic required by next-generation applications and services. To address this problem, Software-defined Networking (SDN) emerges as a management paradigm able to handle these problems through a centralized high-level network approach. However, this centralized characteristic also creates a critical failure spot since the central controller may be targeted by malicious users aiming to impair the network operation. This paper proposes an SDN defense system based on the analysis of single IP flow records, which uses the Gated Recurrent Units (GRU) deep learning method to detect DDoS and intrusion attacks. This direct flow inspection enables faster mitigation responses, minimizing the attack's impact over the SDN. The proposed model is tested against several different machine learning approaches over two public datasets, the CICDDoS 2019 and the CICIDS 2018. Furthermore, a lightweight mitigation approach is presented and evaluated through performance tests regarding each detection method. Finally, a feasibility test is performed regarding the throughput of flows per second that each detection method can analyze. This test is accomplished through the use of real IP Flow data collected at a large-scale network. The results point out promising detection rates and an elevated amount of analyzed flows per second, which makes GRU a feasible approach for the proposed system.

Preprint submitted to Journal of Network and Computer Applications November 12, 2020

Keywords:
Gated Recurrent Units, SDN, Deep Learning, DDoS, Intrusion detection.

1. Introduction

The amount of data traveling on the Internet is rapidly increasing due to the growth in popularity and complexity of connected devices and software solutions. The usage of network resources by end users is rising through the popularization of social networks, web banking applications, and e-commerce, for instance. Thus, new cloud-based services are becoming essential to the operation of this new network environment, which brings specific requirements, such as dynamic traffic allocation (Maenhaut et al., 2017). Furthermore, the increasing popularity of Internet of Things (IoT) devices is gradually changing the Internet scenario by increasing the heterogeneously of communication, since each device (thing) has specific network requirements and processing capability (Yoon and Kim, 2017; Bera et al., 2018). In the face of these changes, management and security are becoming impracticable in traditional static network environments (da Costa et al., 2019; Hajiheidari et al., 2019).

A networking paradigm that is gaining space on several recent pieces of research and applications is the Software-defined Networking (SDN) (Zehra and Shah, 2017; Farris et al., 2019). This network paradigm operates by centralizing the network management into a single programmable controller, able to communicate and control network devices such as switches and routers regardless of their manufacturers, as "white-boxes". The SDN separates the control and data planes so that the central controller is responsible for sending, for instance, management and packet forwarding policies to the controlled devices in a scalable and coordinated manner. This characteristic is a valuable feature able to provide next-generation networks with the dynamic architecture they require (Zhang et al., 2019), in which changes can be performed in a fast, programmable, and on-demand way.

However, while bringing essential improvements to the current network architecture, the SDN, as any centralized service, has as a critical failure spot its central controller. Malicious users may target this controller aiming to impair the whole network operation through the usage of different approaches, such as intrusions (Lopez-Martin et al., 2017) and denial of service (DoS) attacks (Daneshgadeh Çakmakçı et al., 2020; Wang et al., 2020; Xu et al.,

2020; Zhang et al., 2020). Thus, efficient protection mechanisms are needed in SDNs to guarantee the availability of the network and the quality of the provided services (Correa Chica et al., 2020).

The occurrence of these attacks can be generically described as an anomaly, a situation when the network behavior differs from its normal state (Proença et al., 2005). The anomaly detection is a widely approached area, with several different methods proposed in the past years (Fernandes et al., 2019). However, it is still an open research field, since no consensus has been reached due to the enormous amount of different network scenarios and architectures available. In SDN environments, security is a central concern, arousing great interest from the scientific community due to the importance of this paradigm to present and future networks (Maziku et al., 2019).

Among all the anomaly detection methods, the IP flow-based ones are proving to be efficient approaches in SDN environments. It is mainly due to the amount of information these systems can provide, which can be used to characterize the regular network operation with high precision. However, most of the research in this area operates through sampling processes, analyzing the data in intervals of five minutes (Cortez et al., 2006; Bereziński et al., 2015; Shuying Chang et al., 2010), one minute (Pena et al., 2014; Sun et al., 2016), or even in smaller time intervals, such as thirty seconds (Carvalho et al., 2018), and five-seconds (De Assis et al., 2018). While the sampling process helps in scaling the defense system, this process may hide stealthier attacks, such as port scans. Thus, the data analysis performed on each flow separately may provide a more precise detection approach, in which the detection method can find anomalies in specific communications and even identify who is participating in it.

In this paper, we propose a defense system against intrusions and denial-of-service attacks for SDNs based on the analysis of single IP flow records. This individual flow inspection enables faster mitigation responses, ensuring the quality of the services provided by the SDN. The system is divided into two main modules, Detection, and Mitigation.

The Detection Module is responsible for analyzing individual IP flows aiming to identify the occurrence of an anomaly. In this module, we used a recurrent deep learning algorithm called Gated Recurrent Units (GRU) (Cho et al., 2014) as a classifier. Deep learning approaches use multiple layers to learn data representation with various levels of abstraction and is increasingly gaining space among researchers for network applications (Lopez-Martin et al., 2018) (Aldweesh et al., 2020). GRU is widely applied in prob-

lems in which historical information is essential to the performance of classification tasks. In the proposed system, this method inspects individual IP flows through a multidimensional analysis, operating as a binary flow classifier, *i.e.*, classifying them as normal or abnormal.

The Mitigation Module generates efficient counter-measures against the detected attacks. Since the system proposed in this paper individually analyzes IP flows, it can directly identify the attacking node address. Thus, a directed mitigation approach is proposed, which aims to bring the SDN back to its regular operation through a light and straightforward process.

To evaluate the efficiency of GRU as a detection method, we tested it against seven other shallow and deep learning detection approaches over two different scenarios using public datasets. On the first one, named CICDDoS 2019 (Sharafaldin et al., 2019), we tested the methods over several different kinds of Distributed DoS (DDoS) attacks. The second scenario, called CICIDS 2018 (Sharafaldin et al., 2018), was used to test the efficiency of the detection methods against different intrusion techniques. Furthermore, these two datasets are used to measure the proposed mitigation approach's efficiency regarding each one of the evaluated detection methods. The choice of these databases was motivated by their variety of attacks and the number of available IP flow features, an essential characteristic for the application of Deep Learning methods. Finally, we tested the number of flows per second the tested anomaly detection approaches can process to prove the proposed system's feasibility.

We can highlight the following as main contributions of this paper:

- A system for SDN defense against intrusion and DDoS attacks;
- A precise anomaly detection scheme based on isolated IP flow analysis, enabling near real-time detection. This approach allows for faster mitigation responses, minimizing the impact suffered by the SDN;
- The efficiency evaluation and comparison of distinct shallow and deep learning anomaly detection techniques applied in public datasets and the efficiency measurement of the proposed mitigation process.

The remainder of this paper is organized as follows: Section 2 presents state of the art through related work; Section 3 describes the organization of the proposed system; Section 4 details the GRU method used for anomaly detection at the Detection Module; Section 5 discusses the performance outcomes achieved by GRU in comparison with seven other methods and the

performance evaluation of the mitigation approach; Finally, section 6 presents the conclusions and future works.

2. Related Works

Software-defined Networking (SDN) is an emerging paradigm that significantly improves the management procedures, providing the network administrator with the flexibility of dynamic traffic allocation, as well as an online, softwarized, and centralized configuration. Several authors have been developing solutions through the usage of SDNs, such as Theodorou and Mamatas (2017) that demonstrated the operation of CORAL-SDN, an SDN based solution for the Internet of Things. The authors highlighted several benefits from the usage of this paradigm in Wireless Sensor Networks (WSN), such as the centralized control and the elasticity support regarding WSNs requirements.

Although centralized management is one of the main advantages of SDNs, it also represents a weak spot, since the operation impairment of the controller may lead to critical network issues. Thus, the security of this controller is an essential matter to SDN implementation. In Tatang et al. (2017) the authors proposed the SDN-GUARD, a system for detecting and mitigating rootkits in SDN controllers. Their system performs a dual-view comparison to detect malicious programming attempts, and the authors highlighted the achievement of reasonable detection rates with a relatively small performance overhead. In Nam and Kim (2018), the authors addressed the security enhancement of SDNs through the usage of open-source IDS software called Suricata. The authors also described the usage of OpenFlow to implement SDN security mechanisms. In Gkountis et al. (2017), the authors proposed a lightweight DDoS defense algorithm, based on a simple set of rules, in the protection of SDN environments. The authors highlighted that the proposed approach achieved better results in comparison to other legacy protection schemes regarding an SDN ecosystem of mobile users. In Sidki et al. (2016), the authors proposed fault protection for SDN controllers, enabling the network to maintain its regular operation through a redundancy-based approach using a synchronized slave controller.

One of the main approaches for detecting attacks in SDN environments is to characterize the network's normal behavior. Thus, when an abnormal situation is detected, the system may take countermeasures to mitigate the problem. This approach is called anomaly detection, and several different

techniques may perform this task (Pena et al., 2014; Lei, 2017; Qin et al., 2018). On the past years, machine learning (ML) methods have been widely applied as classification systems on the detection of network anomalies. In Nanda et al. (2016), the authors proposed the usage of machine learning algorithms trained over historical data to detect network attacks. They compared the efficiency of four different ML algorithms, the C4.5, Bayesian Network, Decision Table, and Naive-Bayes, achieving around 91% of prediction accuracy through the use of Bayesian Network. In Kornycky et al. (2017), the authors investigated the use of low-cost WLAN dongles to monitor a network, and passively perform traffic classification, improving service monitoring by focusing on enforcing network security policies. To reach this objective, the authors applied different machine learning methods, such as kNN, WkNN, GMM, GMM-UBM, BCT, PTSVQ, and TRAP-VQ. The results point out that TRAP-VQ, a technique proposed by the authors, achieved the highest f-measure among all tested approaches, proving to be efficient for WLAN traffic characterization regarding prior knowledge requirements and computational complexity. In Fukuda et al. (2017), the authors proposed the usage of the Domain Name System (DNS) backscatter as an additional source of information regarding network activity. The authors applied different machine-learning algorithms to classify originator activity of malicious traffic based on the retrieved data, which are classification and regression tree (CART), random forest (RF), and support vector machine (SVM). The proposed algorithm achieved reasonable accuracy and precision outcomes of around 75%, which demonstrates both that DNS backscatter is a good source of network information and that the tested machine learning approaches are efficient for malicious activity detection.

Moreover, several network anomaly detection approaches operate through sampling, analyzing data in time intervals, such as Cortez et al. (2006) and Bereziński et al. (2015), which operates in five-minute ranges. Smaller time intervals implicate on faster anomaly detection, as well as an increase in processing usage for data analysis. In Sun et al. (2016), the authors proposed an anomaly detection method based on Streaming Performance Metrics and Logs operating in one-minute intervals.

Furthermore, a subclass of machine learning algorithms named Deep Learning is increasingly gaining space among researchers in the area (Chowdhury et al., 2019). This subclass describes algorithms that use multiple layers to learn data representation with various abstraction levels, usually when a large dataset is available for training. Deep Learning methods are being

widely applied to detect anomalies in different network environments, such as IoT systems for Smart Cities (Guo et al., 2020) and Industrial control systems (Khan et al., 2020). In Kao and Jiang (2019), the authors proposed an anomaly detection framework for univariate time series. To achieve this goal, they first divide data into three classes, which are stationary, periodic, and non-stationary time series. Then, different statistical and Deep Learning methods, such as GRU, STL, SARIMA, LSTM, LSTM with STL, and ADSaS, are applied over these separated data for performing anomaly detection. The results pointed out that the proposed framework obtained better performance outcomes in comparison to related methods regarding precision, recall, and f-measures. Similarly, in Qin et al. (2018), the authors proposed the usage of Long Short Term Memory (LSTM) networks on the detection of anomalies in IP networks. The outcomes achieved shows promising precision and recall rates, demonstrating the efficiency of the method in classification problems. In Xie et al. (2020), the authors present a network model to predict sensor/controller parameters in industrial control systems using 1D-CNN and GRU. The proposed model was validated through the Secure Water Treatment (SWaT) dataset and achieved promising results regarding precision, recall, and f1-measure metrics. In Qu et al. (2018), the authors proposed a modification of the traditional GRU applied to the identification of user anomalies by analyzing web Logs. The authors' unsupervised approach presented better identification outcomes compared to conventional LSTM and SVM models for anomaly detection. In Liu et al. (2019), the authors present an anomaly detection approach in network logs, using GRU and Support Vector Domain Description (SVDD). Initially, the authors apply the Principal Component Analysis (PCA) method to reduce the databases' dimensionality and extract significant attributes. Subsequently, the processed databases are trained using the GRU-SVDD classification model. The experiments in classical KDD Cup 99 databases showed that the proposed method was more efficient than the classical GRU-MLP and LSTM algorithms.

In this paper, we propose an SDN defense system against DDoS and intrusion attacks. This system, unlike several different approaches described in this section, operates without sampling, acting directly into individual IP flow data, which provides faster detection and, consequently, decreases the impact caused by the attack through a mitigation process. The proposed system performs a multidimensional (multiple flow feature) analysis using GRU deep learning method for detecting attacks on the SDN controller.

3. SDN Defense system

In this section, we describe the operation of the proposed SDN defense system. The management centralization provides many advantages in this paradigm. Still, it is necessary to give the controller security resources to guarantee its operation and, consequently, the quality of the services provided. Thus, in this paper, we propose an SDN security defense system able to detect the occurrence of different intrusion and DDoS attacks on central controllers through an analysis of multi-dimensional IP flows.

Unlike other techniques that analyze traffic data through different time windows (from seconds to minutes), the proposed defense system aims to analyze and identify attacks in individual flows, providing a higher accuracy on the detection and improving the response time for mitigation actions. The main disadvantage of this approach is the amount of data analyzed by the system, which needs to be able to provide fast and accurate classification outcomes. However, it brings the advantages of rapid detection, decreasing the impact caused by the attacks over end users, and users' identification, since IP flow records stores qualitative information, such as source and destination IP addresses and ports used on the communication.

Two different parts compose our SDN defense system, the Detection, and Mitigation modules, which communicate with each other through the central system logic, as observed in Fig. 1. Each module is composed of two other sub-modules, and the SDN system operates within the SDN controller. All data analysis is performed automatically, and the network administrator only receives notifications about detected attack events.

The Detection module is responsible for detecting the occurrence of intrusion and DDoS attacks, as well as generating an alarm that invokes the operation of the mitigation module. In this module, we applied a recurrent Deep Learning approach called Gated Recurrent Units (GRU), which will be detailed in the next section. Since GRU is a supervised learning method, the sub-module called "Training" is responsible for calibrating the classification outcomes through the usage of historical labeled data. The second sub-module is accountable for analyzing flow records and generate a binary result, classifying them as normal or abnormal traffic.

It is essential to highlight that the investigation of different IP flow features (or dimensions) enriches traffic analysis by providing relevant information about the communication and who is participating in it (Shuying Chang et al., 2010). Several different approaches use this characteristic to

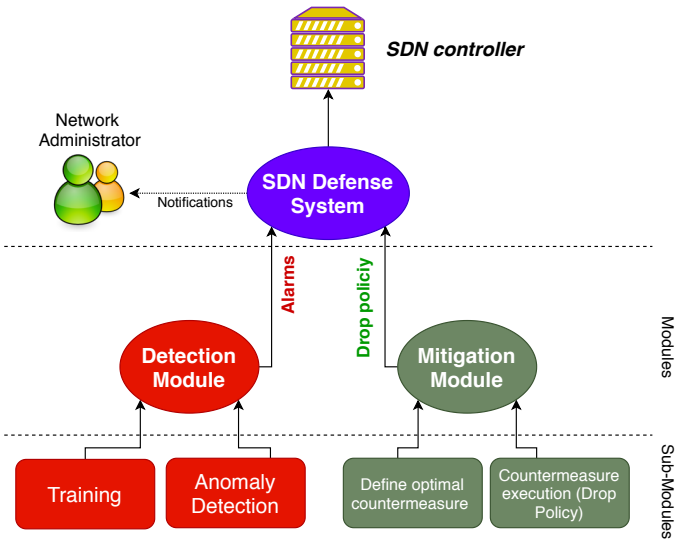


Figure 1: Modular organization of the proposed SDN security system.

improve the performance of anomaly detection (Shuying Chang et al., 2010; Bereziński et al., 2015; Carvalho et al., 2018). Still, most of them use a set of a few features manually selected, such as bits/s and packets/s rates. However, IP flows can provide a much more comprehensive range of information often unused by traditional methods. Unlike most conventional machine learning methods, Deep Learning approaches, such as GRU, can analyze several flow features and automatically give more weight (importance) to those dimensions that most impact the classification outcomes. This feature is indispensable since some patterns may not be obvious, which significantly improves the anomaly detection process.

The Mitigation module is responsible for defining and taking the optimal countermeasures to minimize the attack's impact over the SDN. As its name suggests, the first sub-module determines the optimal countermeasure against the detected anomaly. In contrast, the second one sends the optimal drop policy to the SDN controller for implementation.

Different techniques may be used on the "Define optimal countermeasure" sub-module to define the better mitigation action against the detected attack. Since the proposed defense system analyzes single IP flows, it is possible to directly identify the attacker node, individually discarding related flows

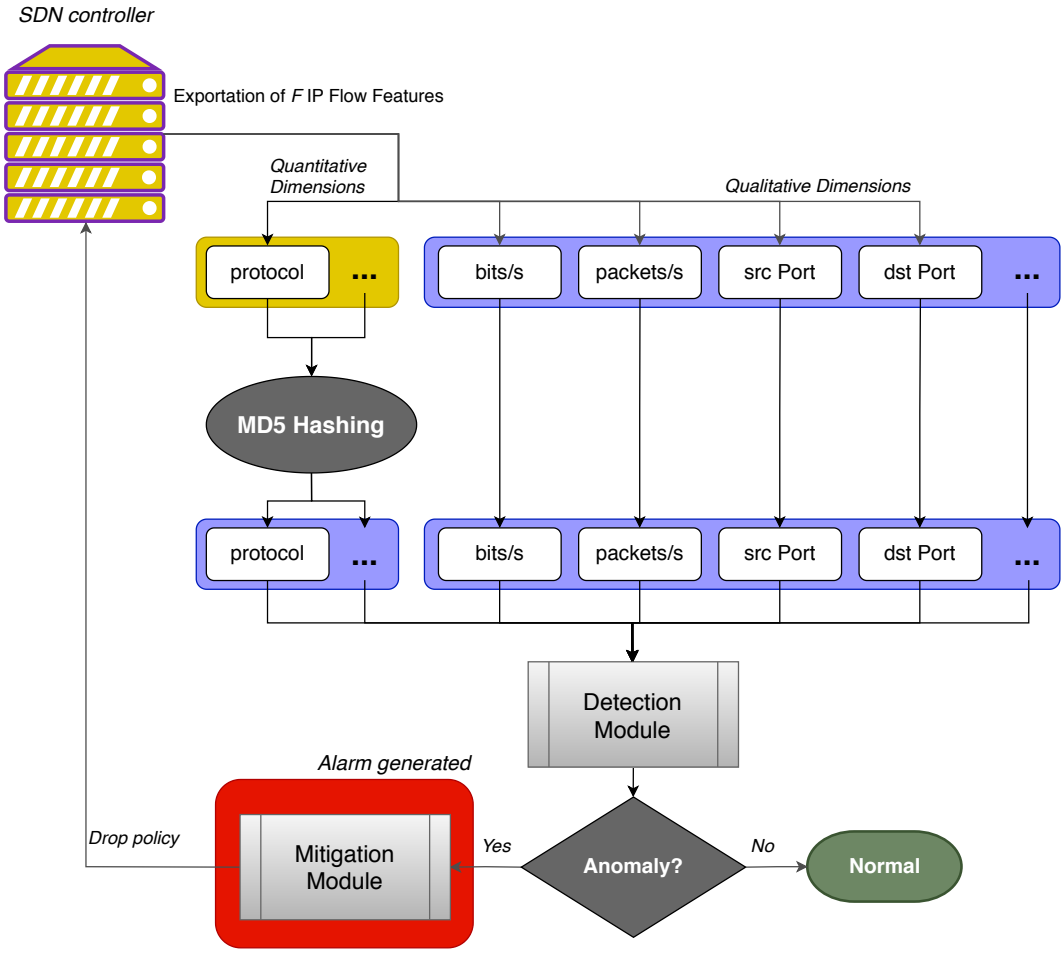


Figure 2: Overall operation of the SDN security system.

communication. Thus, there is no need for a probabilistic drop estimation, which could increase the system's operation's overall computational cost.

Therefore, we propose a directed mitigation approach, which represents a straightforward and light drop schema. Using the information provided by the detection module, such as source IP address, protocol, and destination IP address and port, the system generates an individual drop policy against the attacker IP. This policy is implemented by the SDN controller as soon as the first abnormal flow is detected. Hence, this mitigation schema's efficiency is directly related to the efficiency of the attack detection (method).

Fig. 2 summarizes the operation of the proposed SDN defense system.

As observed, the SDN controller export single IP flow records composed of F features or dimensions. Most of these features are quantitative dimensions that can be directly analyzed by the detection method. However, if the record contains qualitative dimensions, such as the "protocol" element, this data is submitted to an MD5 hashing process to convert them into quantitative values. An essential part of this treatment is to not include the features source IP address and port, and destination IP address on the anomaly detection step. Their usage could compel the detection method to learn patterns that decrease its generalization capacity, stating that, for instance, DDoS attacks always aim at a specific IP address.

After this step, the IP flow record is submitted to the Detection module, in which it will be presented to a binary classification performed by the GRU method. If no anomaly is detected, then the analysis starts all over again with the evaluation of the next flow record. Otherwise, an alarm is triggered, and the Mitigation module is invoked to generate the optimal drop policy against the detected attack. This policy is transmitted to the SDN controller for logic implementation, which forwards the mitigation messages to the network's routers and switches, securing the SDN environment. Fig. 3 summarizes the drop policy generation through the mitigation approach proposed in this paper.

As presented by Fig. 3, the Mitigation Module receives the flow record in which the attack was detected (1). After, it extracts relevant information to generate the drop policy. Since this paper introduces individual flow records analysis, it is possible to identify the attacker node's IP address directly. Thus, this IP address is extracted (2), and a drop policy is generated. Finally, this drop policy is sent to the SDN controller (3) for implementation.

It is essential to highlight that the defense system, unlike traditional approaches, operates autonomously, from the flow extraction to the attack's detection and mitigation. This characteristic is a crucial feature to guarantee a fast response regarding the mitigation process, aiming to minimize the impact caused by the attack. When an attack is detected, an alarm is triggered and automatically invokes the mitigation module. This alarm is also received by the network administrator, but only as a warning notification, as described in Fig. 1.

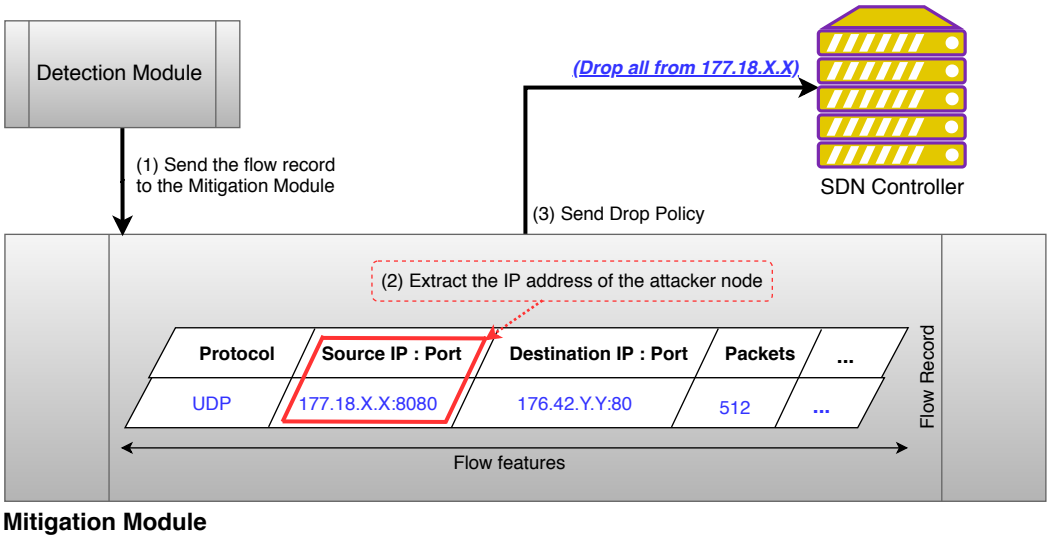


Figure 3: Illustration of the system's drop policy generation using a directed mitigation approach (individual flow analysis).

4. Gated Recurrent Units to detect network attacks

Deep learning methods are becoming increasingly popular among researchers through its usage in various applications aiming to detect computer network attacks and anomalies. Deep learning is a class of machine learning that can retrieve patterns in complex data and, therefore, is widely applied to problems of image recognition, pattern classification, and time series prediction (McDermott et al., 2018). Deep learning stands for the concept of successive layers of representations, as its depth is known as the number of layers representing a model. Deep learning approaches typically use three or more layers of representation, whereas "shallow" learning models, such as Multi-Layered Perceptron (MLP), operates through only one or two layers of learning.

One of the most significant benefits of deep learning methods is the absence of manual feature engineering (McDermott et al., 2018). Therefore, there is no need for prior feature selection, as these methods can automatically find patterns among massive datasets during the training step. These patterns are identified, for instance, by weight matrices, which are used to give more "importance" to features that most impact the classification process. Since IP flow protocols provide a wide range of different dimensions

to describe network communications, some elaborate attack patterns, sometimes less evident to human eyes, can be extracted from the dataset, which significantly improves the classification outcomes.

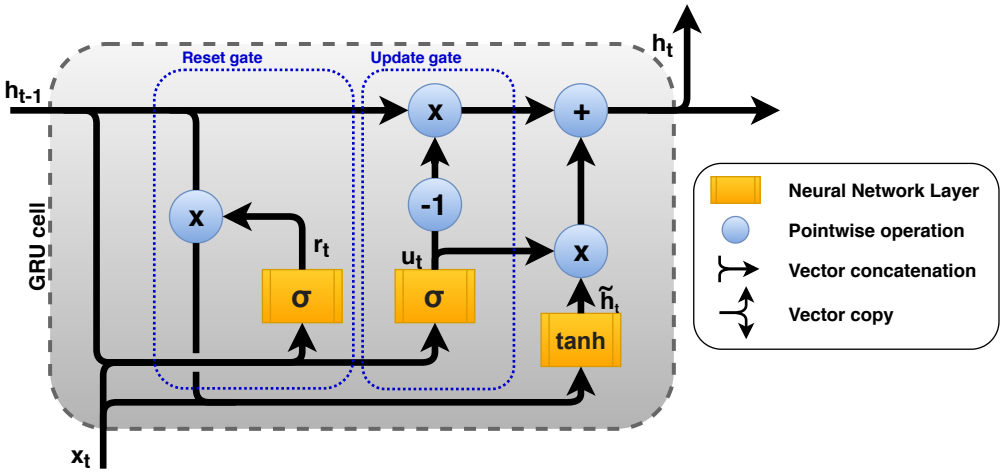


Figure 4: Representation of a GRU cell.

Among the different deep learning approaches, the recurrent ones are promising on the detection of network anomalies and attacks. Unlike the so-called feedforward networks, such as densely connected and convolutional networks, Recurrent Neural Networks (RNN) have memory, *i.e.*, considers past information on prediction/classification process (McDermott et al., 2018). This memory is an essential feature to detecting anomalies, since the state of the network before the occurrence of an attack (previously analyzed IP flows) may be used as well as the current analyzed flow behavior itself to generate alarms. In short, RNNs handle sequences by iterating through the elements of the series and maintaining a state that contains information about what it has seen so far Cho et al. (2014).

However, RNNs have an issue known as vanishing gradient problem. Although this network should theoretically be able to retain information about inputs seen many timesteps before, in practice, RNN are unable to learn long-term dependencies (He and Droppo, 2016). In short, this occurs because successive operations in long-term data gradually reduce their significance and, thus, the more in-depth the analysis, the less meaningful these data are to influence the network outcome (Bengio et al., 1994).

Different approaches have been proposed to address this problem, while

the most commonly used in literature is the Long-Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997). This method implements a series of mechanisms called gates, which can regulate learning and forgetfulness rates, guaranteeing that long-term data maintains its influence over recent predictions. Recently, Cho et al. (2014) proposed a modified version of the LSTM called Gated Recurrent Units (GRU), which summarizes the operation of LSTMs by reducing the number of gates while maintaining the relevance of long-term memories (Cho et al., 2014). The efficiency of LSTM and GRU differs regarding the application they are being applied to, as both were not significantly different in classification accuracy (Zhang et al., 2018). However, GRU has fewer tensor operations in comparison to LSTM, which makes its training process faster.

In this paper, we propose the usage of the GRU method for detecting DDoS and intrusion attacks over SDN environments. This method, as well as LSTM, operates through the utilization of gates that are different neural networks that decide which information should be forgotten or retained. GRUs works using two gates, the Update and Reset ones. The first one is responsible for defining what information regarding a new entry will be forgotten and what new information will be added. In contrast, the second one describes how much long-term or past data will be forgotten. Fig. 4 represents the operation of a GRU cell and its gates.

As presented by Fig. 4, h_{t-1} stands for the hidden state of time interval $t-1$, while x_t and h_t represents input data and output hidden state on the current interval t, respectively.

As previously stated, gates are different neural networks used to define which information to forget or retain. As shown in Fig. 4, both gates operate through a sigmoidal activation function, which is applied to simplify this process since it normalizes its output in values between 0 and 1. Thus, any value multiplied by 0 will be forgotten, while values multiplied by 1 will be kept.

To compute the value of h_t, the cell starts by concatenating h_{t-1} and x_t, and then submitting the resulting vector to the Reset and Update gates, obtaining their output r_t and u_t through Eq. (1) and (2), respectively.

$$r_t = \sigma(W_r \ \cdot [h_{t-1}, x_t] + b_r) \tag{1}$$

$$u_t = \sigma(W_u \ \cdot [h_{t-1}, x_t] + b_u), \tag{2}$$

where W_r and W_u stand for the weight matrices of the neural networks, while b_r and b_u are the neural networks' bias vector. Then, a pointwise multiply is performed between r_t and h_{t-1}, and the result is concatenated with x_t and submitted to a third neural network, with a Hyperbolic Tangent ($tanh$) activation function this time. The use of $tanh$ normalize data between -1 and 1, regulating the output of the neural network and preventing data from being over or undersized between iterations. The output $\tilde{h_t}$ of this neural network is computed through Eq. (3).

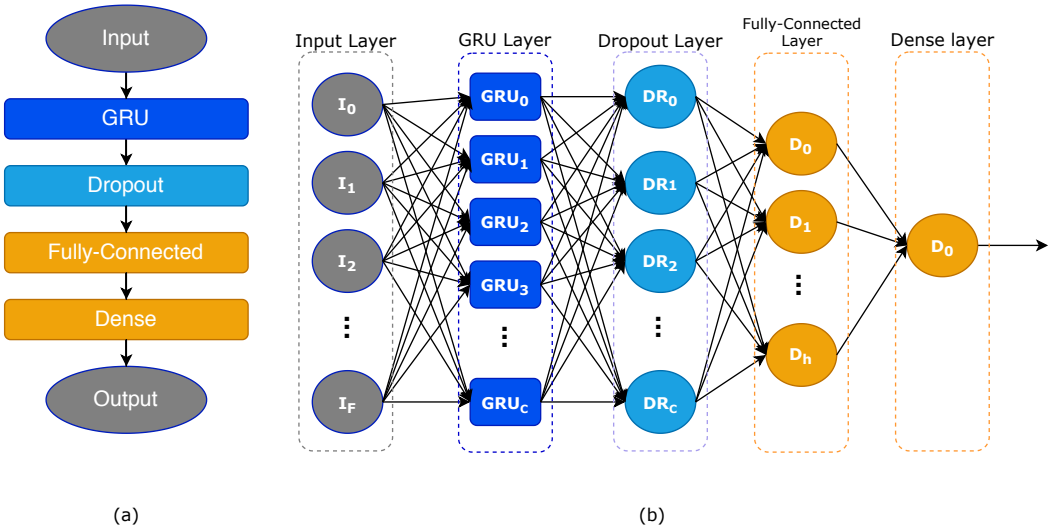


Figure 5: GRU architecture through a high-level representation (a), and through a traditional neural network representation (b).

$$\tilde{h_t} = tanh(W_o \cdot [r_t * h_{t-1}, x_t] + b_o), \tag{3}$$

where W_o and b_o stand for the weight matrix and bias vector of the neural network, respectively. The outcome of the Update gate u_t is used for two situations. On the first one, it decides which part of the new information to add by multiplying its outcome with $\tilde{h_t}$. In the second one, it determines which data to throw away by pointwise multiplying h_{t-1} with $1 - u_t$. Finally, a pointwise addition is performed between these two outputs, generating the GRU cell's result, the hidden state h_t. Eq. (4) describes this situation.

$$h_t = (1 - u_t) * h_{t-1} + u_t * \tilde{h_t} \tag{4}$$

As this process describes the operation of a single GRU cell, the depth of this network is represented by the number of cells C used to fit the classification regarding the stated problem.

The configuration of the GRU implemented in this paper is described in Fig. 5.

The architecture of the network is composed of a GRU layer ($C = 32$), followed by a Dropout layer (drop rate of 0.5), added to save the system from overfitting, and a Fully-Connected layer ($h = 10$ neurons) that performs a global model classification. The output is given by a single neuron (Dense layer) with a sigmoid activation function for binary classification, *i.e.*, classifying the flow as legitimate or malicious. The parameter estimation of the referred values was defined through extensive empirical testing.

Figs. 6 and 7 present the result of the tests that supported the choice of GRU network parameters, which are the number of GRU cells and the number of neurons on the fully-connected layer. Fig. 6 tests different quantities of GRU cells regarding the accuracy and number of analyzed flows per second. This test was performed through the use of the CICDDoS 2019 dataset. As observed, the more cells are added to the GRU layer, the better are the classification results. However, it is essential to consider the number of flows per second the model can classify since the proposed system aims to analyze individual records. As shown by this figure, GRU achieves a balanced tradeoff between accuracy and flow throughput with 32 cells, presenting an efficient and feasible outcome.

Fig. 7 compares the usage of different amounts of neurons at the fully connected layer, also examining the efficiency of GRU regarding accuracy and IP flow throughput. This test was performed using the CICIDS 2018 dataset, as the results achieved using the CICDDoS 2019 dataset are similar to each other. This graph shows that this layer does not considerably influence the number of analyzed flows/s that the neural network can analyze. Furthermore, the lowest accuracy rate was achieved with 0 neurons, *i.e.*, without adding the fully-connected layer before the output one. Since the accuracy levels seem to achieve similar outcomes with 10 or more neurons, we set $h = 10$ neurons to reduce computational cost.

The drop rate at the Dropout layer was chosen based on Srivastava et al. (2014), in which the authors state that 0.5 appears to be near to optimal for most networks and tasks.

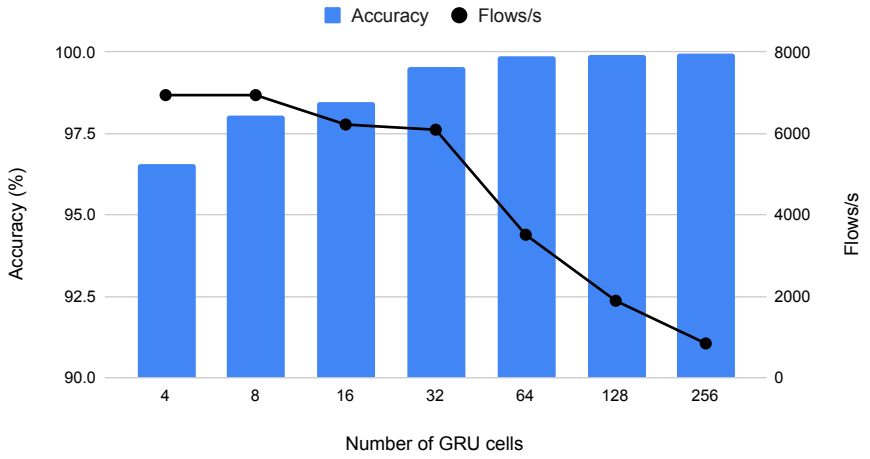


Figure 6: Comparison of different numbers of GRU cells regarding accuracy and amount of analyzed flows per second through the CICDDoS 2019 dataset.

5. Tests and results

This section analyzes the performance results relating to the detection and mitigation modules for the security system. We have applied the GRU method together with the directed mitigation approach. Aiming this objective, we compared the proposed method with the following detection approaches: Deep Neural Network (DNN) (Abdulhammed et al., 2019), Convolutional Neural Network (CNN) (Kwon et al., 2018), Long-Short Term Memory (LSTM) (Qin et al., 2018), Support Vector Machine (SVM) (Lei, 2017), Logistic Regression (LR) (Yadav and Selvakumar, 2015), k-Nearest Neighbors (kNN) (Divyatmika and Sreekesh, 2016) and Gradient Descent (GD) (Wijnhoven and de With, 2010). These methods were selected based on related works, as they represent efficient and feasible approaches for application environments similar to the one proposed in this work.

All methods were implemented using Python, Keras (GRU, LSTM, CNN, and DNN), and Sklearn (SVM, LR, kNN, and GD), on a computer using Windows 10 64bit, Intel Core i7 2.8GHz, and 8GB of RAM. DNN was implemented using 3 hidden layers, composed of 100, 40, and 10 neurons. CNN was implemented using 3 layers, with 64, 32, and 16 filters with kernel size of 16, 8, and 3, respectively. LSTM was configurated with 32 units, similarly to our GRU implementation. The SVM method was configured with a linear

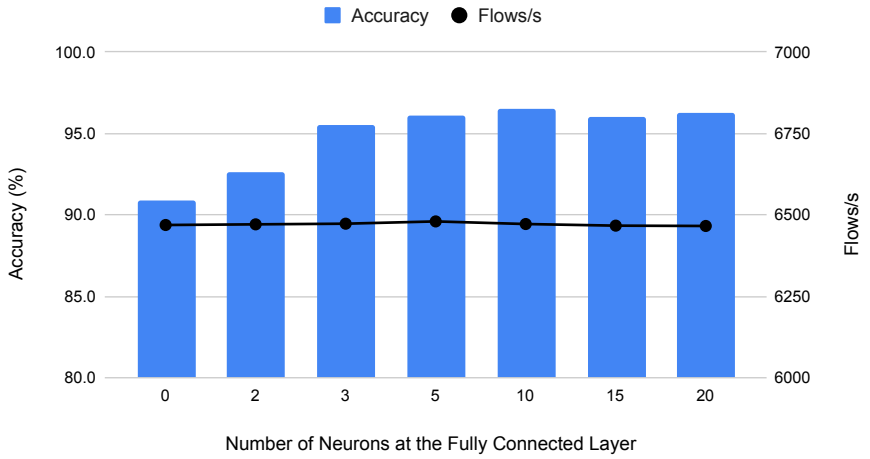


Figure 7: Comparison of different numbers of neurons at the fully-connected layer regarding accuracy and amount of analyzed flows per second through the CICIDS 2018 dataset.

kernel. kNN method, in turn, was defined with the number of neighbors equals 3. Finally, LR and GD were configurated with Sklearn default values. All methods were tested through 100 epochs, as many methods converge with fewer iterations.

To compare the efficiency of the tested methods, we applied traditional classification metrics, such as accuracy, precision, recall, and f-measure. The accuracy presents the percentage of correctly classified flow records. Precision is used to measure the ratio of IP flows correctly recognized as abnormal among all the samples classified as unusual. The recall metric estimates the percentage of correctness for anomalous flows. F-measure represents the harmonic mean between true-positive rate (malicious flows classified as anomalous) and precision.

Two different test scenarios were used in this performance analysis, both of them using public datasets. Furthermore, we tested the number of flows per second the anomaly detection approaches can process since it is a vital feature for the system's operation. In the next sections, we describe each test scenario and present the results achieved by the tested methods on them.

5.1. Scenario 1 - CICDDoS 2019 Dataset

We applied simulated IP flows collected from the public dataset CICDDoS 2019 (Sharafaldin et al., 2019). The authors generate realistic background

traffic profiled through B-Profile System (Sharafaldin et al., 2017) to abstract the behavior of human communications for legitimate traffic through different protocols, such as HTTP, FTP, and SSH. In this dataset, the authors simulate the behavior of 25 users.

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

This dataset provides data with $F = 87$ extracted IP Flow features, such as source and destination IP addresses and ports, protocols, several flags, counters, and flow identification features. However, we used only 83 features since the dimensions "source and destination IP address," "source port" and "Flow ID" was not used for training to avoid data bias. The "destination port" feature was maintained since several network applications operate through a default port, which could help the detection of attacks at specific servers. The remaining data was submitted to a formatting process to convert qualitative features into quantitative ones like described in section 3.

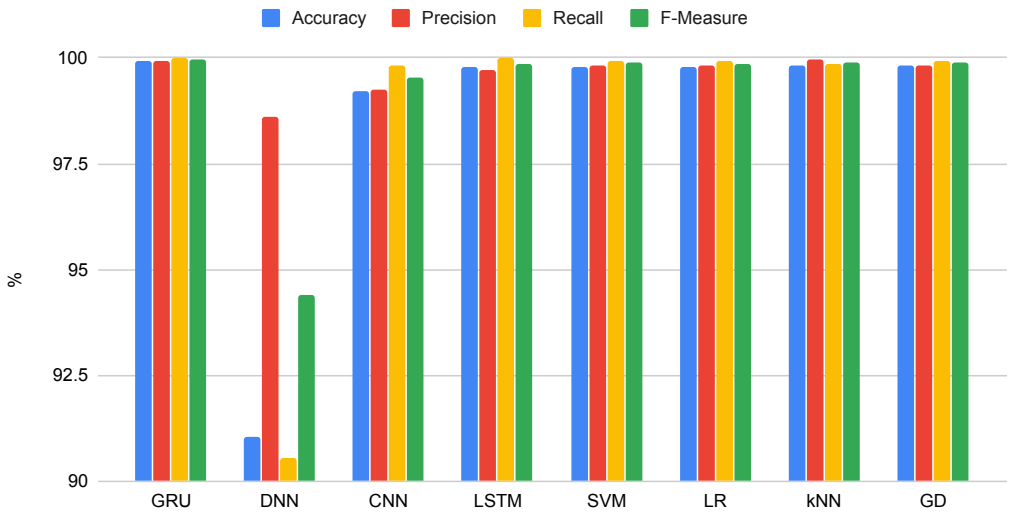


Figure 8: Individual results for each tested method regarding accuracy, precision, recall and f-measure rates - first test scenario.

Fig. 8 presents the results achieved by each method, in which, for each one, we analyze the accuracy, precision, recall, and f-measure separately.

As observed, most tested methods achieved excellent outcomes, with metrics' results near 100%. Even DNN, which fared worse than the other tested methods, achieved an accuracy rate of around 91%. CNN method produced better results compared to DNN but is also visually worse than the different tested approaches. The results' similarity achieved by the remaining techniques is mainly due to the characteristics of the attacks used in this scenario. Although there are some differences regarding DDoS types, they are primarily flooding attacks, a behavioral pattern that the tested methods seem to identify efficiently. Fig. 9 shows the average of the metrics' outcomes achieved by each technique. These results summarize the tested approaches' performance outcomes into a single metric, facilitating the visualization of which one fared better.

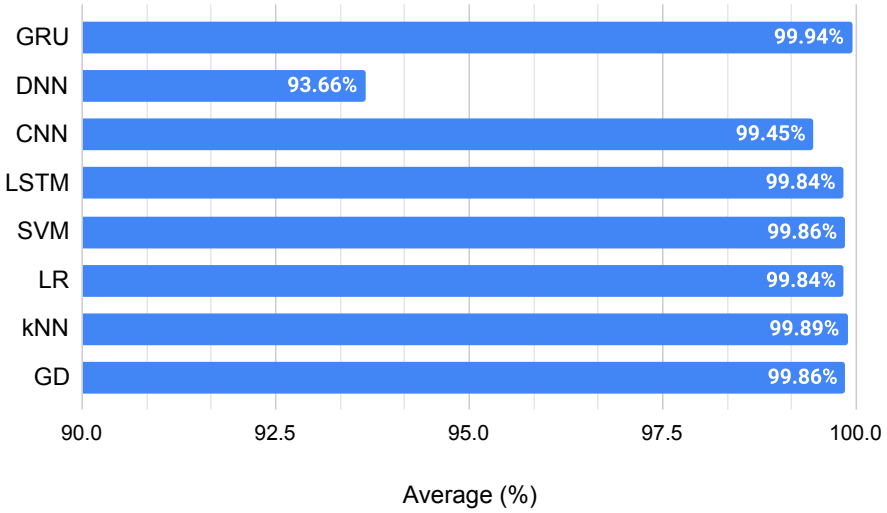


Figure 9: Average results for each tested method regarding accuracy, precision, recall and f-measure rates - first test scenario.

Through the analysis of this figure, it is possible to infer that the outcomes achieved by GRU, LSTM, SVM, LR, kNN, and GD are equally efficient in this test scenario since the difference between them is minimal. The GRU method fared slightly better than the other approaches, achieving one of the most balanced outcomes between the four tested metrics.

Aiming to measure the differences between the tested methods further, we performed a separate analysis of the data, measuring their effectiveness on classifying normal (specificity) and attack flows (recall or sensitivity). The results are presented in Fig. 10.

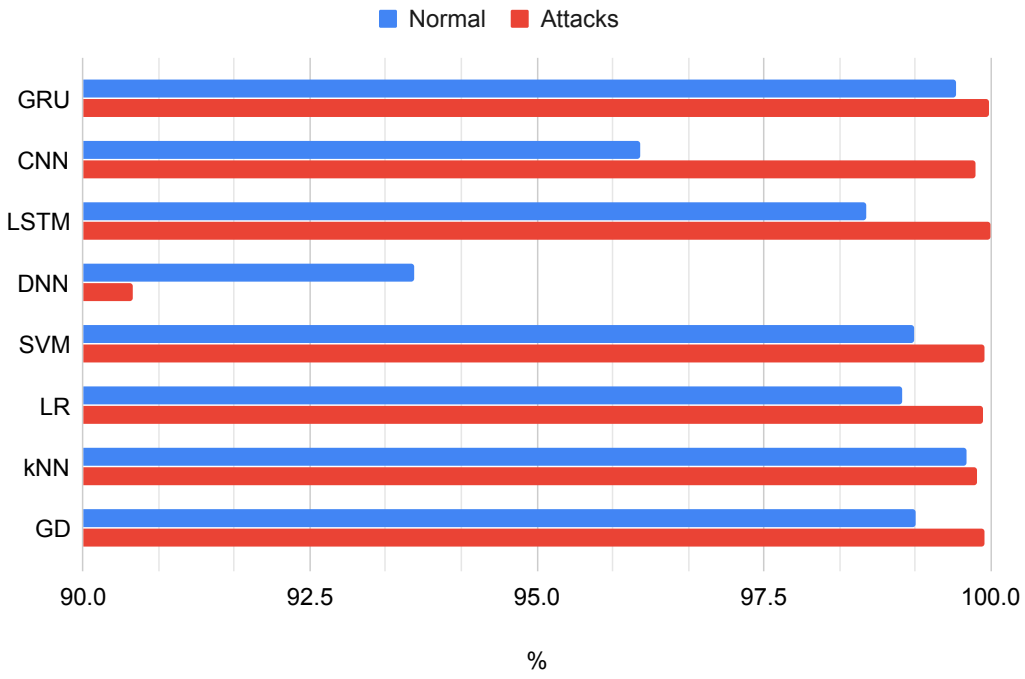


Figure 10: Proportion of correctly identified normal (specificity) and attack (recall) IP flows of each tested method - first test scenario.

As observed in Fig. 10, most of the methods achieved similar results on classifying attack flows, differing mostly on the classification of the normal ones. In this test scenario, kNN and GRU achieved the best outcomes on legitimate flow classification, with rates of 99.7% and 99.6%, respectively. They are followed by SVM, GD, LR, LSTM, CNN, and DNN, which achieved specificity rates of 99.1%, 99.1%, 99.0%, 98.6%, 96.1%, and 93.6%, respectively. Although these results appear to have a small difference between each other, this difference becomes more significant as the analyzed network size increases, which directly influences the mitigation efficiency.

5.2. Scenario 2 - CICIDS 2018 Dataset

In this test scenario, we applied simulated IP flows collected from the public dataset CICIDS 2018 Sharafaldin et al. (2018). As described in the previous test scenario, the authors also generate realistic background traffic profiled through B-Profile System (Sharafaldin et al., 2017) to describe human communications for legitimate traffic. In this dataset, the authors simulate the behavior of 500 machines, 420 legitimate users, 30 server nodes, and 50 attacking nodes. The authors used a system called M-Profile to generate different attack scenarios in which the devices operate specific tasks accordingly to the attack's type. They are:

- Infiltration of the network from inside;
- HTTP denial of service;
- Collection of web application attacks;
- Brute force attacks;
- Last updated attacks.

Unlike the previously addressed dataset, the CICIDS 2018 does not divide the data into training and test groups. Thus, we randomly selected 66% of the dataset for training and used the remaining 34% for testing.

This dataset provides data in two different extensions: ".pcap", the standard extension for flow export, and ".csv", in which the authors present data ready for training on machine learning methods, *i.e.*, without qualitative dimensions such as source and destination IP addresses and Flow ID. We used the ".csv" files and, thus, in this scenario, the methods operate through the usage of $F = 79$ features provided.

This test scenario has a more significant amount of computers performing legitimate communications. Furthermore, intrusion attacks need to cause minimal impact on the network's behavior to avoid detection, unlike flooding attacks that aim to impair network operations. These characteristics make this test scenario a challenge for the anomaly detection approaches.

Fig. 11 presents the individual results achieved by the tested methods.

As observed, for this scenario, the methods' results for the test metrics are more heterogeneous. However, except for DNN, all approaches achieved

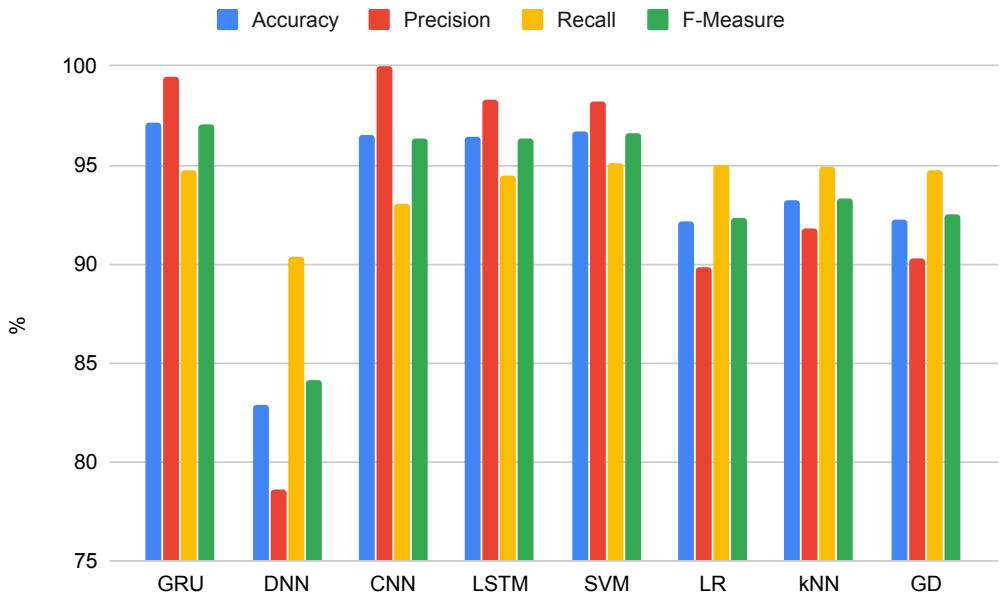


Figure 11: Individual results for each tested method regarding accuracy, precision, recall and f-measure rates - second test scenario.

accuracy rates higher than 90%, with GRU (97.1%), CNN (96.5%), LSTM (96.4%) and SVM (96.6%) achieving similar results.

For the precision metric, the CNN method fared better than the other methods, with a rate of 99.9%, followed by GRU, with a 99.4% rate. LSTM and SVM methods achieved the similar precision rates of 98.3% and 98.2%, respectively, followed by kNN (91.7%), GD (90.3%), LR (89.8%) and DNN (78.6%).

For the recall metric the SVM method achieved the best outcomes with a 95.1% rate, closely followed by LR and kNN, with rates of 95% and 94.9%, respectively. GD and GRU achieved similar results of 94.8% and 94.7%, respectively, and LSTM fared slightly worse than the already cited methods in this metric, with a 94.4% rate. CNN and DNN fared worse, reaching recall rates of 93% and 90.4%, respectively.

Finally, for the f-measure, GRU achieved better outcomes, with a 97% rate, being closely followed by SVM (96.6%), CNN (96.4%), and LSTM (96.3%). kNN, GD, and LR methods achieved similar results for this metric, with rates of 93.3%, 92.5%, and 92.3%, respectively, while DNN fared worse

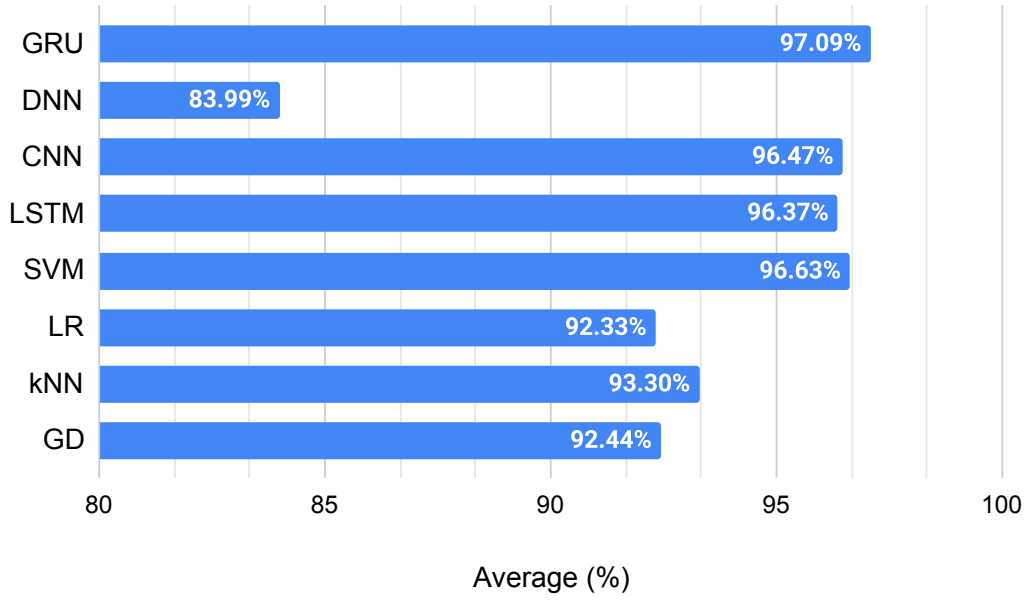


Figure 12: Average results for each tested method regarding accuracy, precision, recall and f-measure rates - second test scenario.

with a value of 84%.

Similarly to the first scenario, Fig. 12 presents the average results regarding the four tested metrics, performing an overall analysis on which method fared better.

As observed, GRU achieved better overall results, being the most balanced approach regarding the accuracy, precision, recall, and f-measure results. It is followed by SVM, CNN, LSTM, which achieved similarly reasonable overall rates. Finally, kNN, GD, LR, and DNN fared worse in comparison to the other tested approaches.

Similarly to the previously analyzed test scenario, we evaluate the effectiveness of the methods' classification regarding normal (specificity) and attack (recall or sensitivity) flows separately. The results are presented in Fig. 13.

As shown in Fig. 13, once again, most of the methods presented similar outcomes regarding the detection of attack flows, while the results differ regarding the correct classification of legitimate flows. In this test scenario, GRU and LSTM achieved the best classification results of normal flows, with

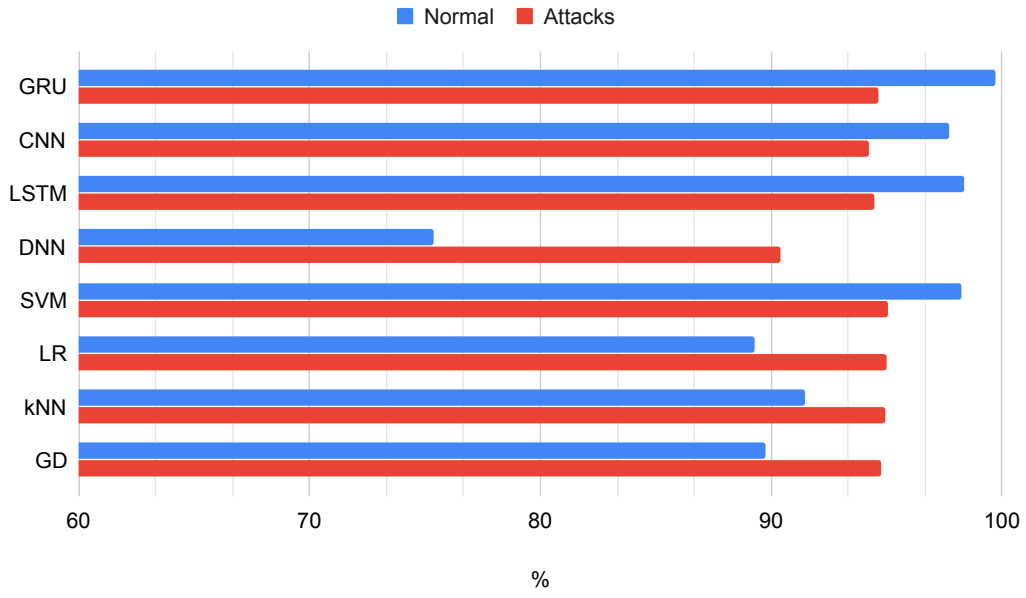


Figure 13: Proportion of correctly identified normal (specificity) and attack (recall) IP flows of each tested method - second test scenario.

rates of 99.7% and 98.3%, respectively. They are followed by SVM, CNN, kNN, GD, LR, and DNN, which achieved specificity rates of 98.2%, 97.7%, 91.4%, 89.8%, 89.2%, and 75.3%, respectively.

We believe GRU fared better than the other tested methods because of its ability to learn long-term dependencies. This characteristic improves the classification of both normal and abnormal flows, generating more balanced outcomes regarding precision and recall rates, unlike the other tested approaches. However, this ability is also present on the LSTM method, which achieved inferior results in comparison to GRU in both scenarios. Accordingly to Jozefowicz et al. (2015), which performed empirical evaluations to compare both approaches, it is not clear which one present the best results. Although both methods tend to generate similar results, one of them can be more efficient than the other depending on the application scenario (which includes dataset size, number of analyzed features, and so on). GRU is a more straightforward method, which enables a faster training process in comparison to LSTM, which has a more complex structure to learn and describe traffic behaviors. Even though LSTM should achieve better results in more

massive datasets due to this characteristic, we can conclude that for these test scenarios, the usage of GRU is more efficient in detecting DDoS and intrusion attacks.

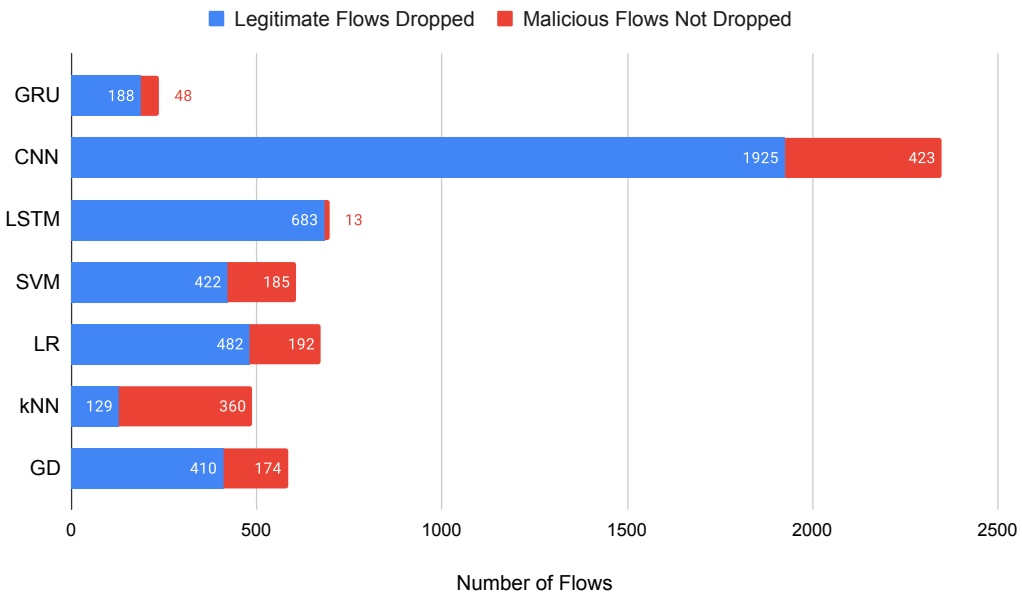


Figure 14: Number of legitimate flows dropped and malicious flows not dropped for each tested method, regarding the dataset CICDDoS 2019

5.3. Mitigation outcomes

As this paper proposes the analysis of individual IP flows for attack detection, it is possible to directly identify which hosts are participating in the communication process and determine the attacker node. Therefore, a simple, straightforward mitigation approach is presented, which drops the identified target packets as soon as the detection occurs.

Thus, the efficiency of the mitigation is directly influenced by the detection approach. To evaluate the mitigation outcomes for each detection method tested, we analyze two metrics: i) the absolute number of normal (legitimate) flows dropped, and ii) the absolute number of attack (malicious) flows not dropped. These values are stacked to summarize the mitigation approach's efficiency in each tested detection method. Fig. 14 and 15 present

the mitigation results relating the datasets CICDDoS 2019 and CICIDS 2018, respectively.

As observed in Fig. 14, although the kNN method dropped a lesser amount of legitimate flows in this test scenario, it could not detect as many DDoS intervals than GRU. Thus, GRU achieved the most balanced outcome, guaranteeing both the detection of malicious flows and preserving legitimate ones. Although the LSTM method achieved an outstanding mitigation rate regarding attack flows, it was less efficient than GRU on detecting normal ones.

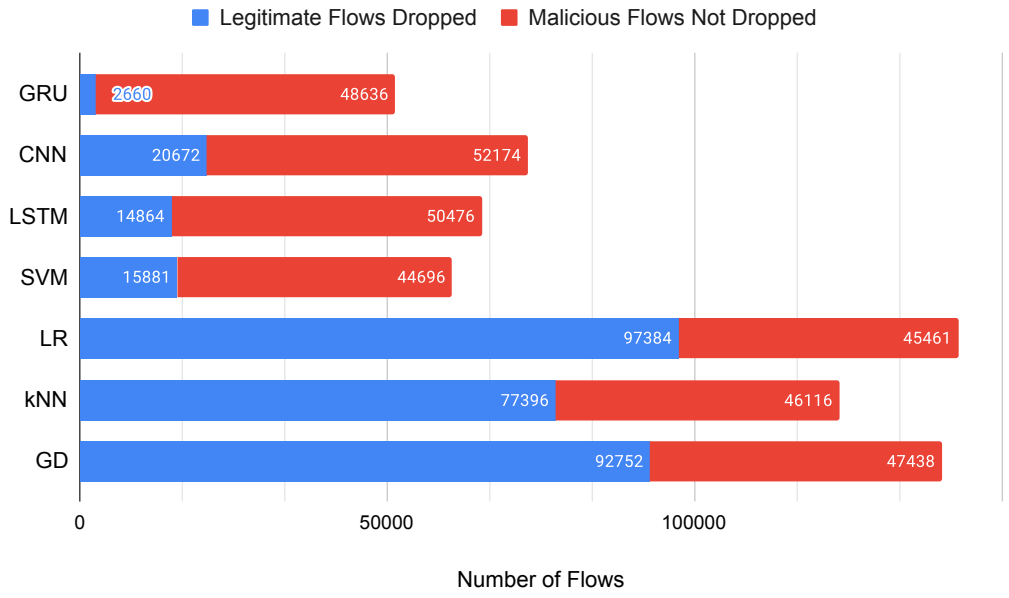


Figure 15: Number of legitimate flows dropped and malicious flows not dropped for each tested method, regarding the dataset CICIDS 2018.

As shown in Fig. 15, GRU once more achieved the most balanced outcomes in this test scenario. Although the amount of malicious flows not dropped by this method is more significant than the ones conducted by SVM, LR, kNN, and GD, it correctly identified most of the legitimate flows, reducing the impact caused to regular users.

The difference between the values presented by Fig. 14 and 15 occurs due to the size of the test datasets, which are summarized by Table 1.

Table 1: Amount of flows available for testing regarding both datasets CICDDoS 2019 and CICIDS 2018.

	Legitimate flows	**Attack flows**	**Total**
CICDDoS 2019 (testing day)	49,763	248,815	298,578
CICIDS 2018 (33%)	906,094	907,742	1,813,836

As observed, this difference occurs due to the number of flows available for testing, since the CICIDS 2018 testing set was six times bigger than the CICDDoS 2019 one. Furthermore, intrusion attacks tend to be stealthier than flooding ones, which elevates the complexity of detection. These values were presented in an absolute form to illustrate how the network's size influences the mitigation outcomes, especially regarding the impact on legitimate users. Therefore, the larger the SDN network (regarding the throughput of IP flows), the more evident is the difference in detection of the evaluated methods.

Since this mitigation approach directly depends on the detection method's efficiency, it represents a low cost, fast mitigation process. Although this approach proved to be efficient, it may replicate misclassifications through time. The generation of a drop time window may improve the mitigation outcomes, which should be implemented in future works.

5.4. Feasibility of implementation

Since the proposed SDN defense system aims to inspect the traffic data through individual flow analysis, the detection method should be both lightweight and efficient. These characteristics are essential, since delays on the detection may impair the operation of the mitigation approach, as well as cause more damage to the SDN.

To measure the efficiency of the presented approach, we calculate the average number of flows per second the tested anomaly detection methods can analyze and classify. The rates were collected through 10 different executions, and the results are shown in Table 2. The standard deviation of the calculated average was added to illustrate that, in the ten executions, the results achieved showed little variation. Thus, it can be noted that there was no bias in the results due to external interference, such as running other programs, for example.

As observed, the LR and GD methods are by far the fastest, in comparison to the other methods. However, their results regarding the classification

Table 2: Average amount of flows/s each method is able to process.

	Average of Flows/s	Standard Deviation
GRU	6,469	0.044
DNN	87,561	0.008
CNN	18,318	0.408
LSTM	5,298	0.272
SVM	1,470	0.661
LR	7,344,471	0.002
kNN	1,597	1.013
GD	7,307,621	0.002

performance were among the worst tested in this paper. They are followed by DNN, which can classify 87561 flow records per second, CNN, GRU, LSTM, kNN, and SVM.

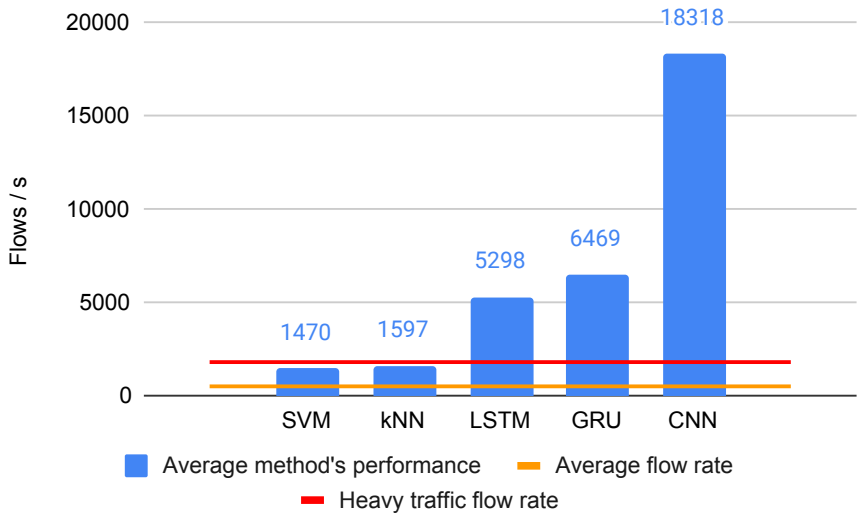


Figure 16: Flow/s rates of GRU, CNN, LSTM, SVM, and kNN methods in comparison to real data rates collected form a real-world, large-scale network.

To verify if the tested methods' results are feasible in real-world scenarios, we collected real IP flow data from the State University of Londrina (Brazil) (Orion, 2020), a large-scale network composed of about 7000 different active hosts. The data was collected for a whole week through intervals of one

second, and, for each day, we measured the average flow/s rate. On regular days, around 500 flows/s pass through the collector, while on heavy traffic days, they achieved peaks of a maximum of 1681 flows/s. Fig. 16 presents the average flow/s rates reached by the tested methods compared to the average and worst-case rates observed through the real-world environment data. For ease of data visualization, we removed the results of DNN, LR, and GD from the graph. Although these methods achieved a much higher flow/s rate compared to the others, they presented inferior outcomes regarding attack detection. Thus, Fig. 16 shows the comparison of the most efficient detection methods.

By comparing the achieved results with the real collected data, we conclude that GRU is a feasible method for real-world anomaly detection. In this case study, it was able to analyze around two times the amount of flows/s required in dense traffic situations, operating within the limited CPU and memory resources of a personal computer. Furthermore, it presented the most balanced results regarding the accuracy, precision, recall, and f-measure for both test scenarios. Thus, it is possible to conclude that GRU is a promising method to operate within the Detection module of the proposed SDN defense system.

6. Conclusions and Future Work

In this paper, we proposed an SDN defense system against intrusion and DDoS attacks. This approach can protect the SDN central controller against situations that may compromise it, consequently impairing the network operation. The proposed system is composed of two main parts, the Detection and Mitigation modules. The Detection module is responsible for detecting the occurrence of attacks, while the Mitigation module takes the required countermeasures to reduce its impact over the network and, consequently, its users. The proposed approach individually analyzes and classifies IP flows into normal or abnormal, which enables a faster detection process that contributes to minimizing the attack's influence over the SDN controller. Furthermore, this individualized analysis of IP flows can easily identify attackers and their targets through feature extraction.

We also propose the usage of the Gated Recurrent Units (GRU) method on the system's Detection module. GRU is a recurrent deep learning approach that simplifies the operation of the Long-Short Term Memory (LSTM) method while maintaining similar performance outcomes. To test the ef-

ficiency of the proposed method, we compare it with seven different approaches. They are Dense Neural Network (DNN), Convolutional Neural Network (CNN), Long-Short Term Memory (LSTM), Support Vector Machine (SVM), Logistic Regression (LR), k-Nearest Neighbors (kNN), and Gradient Descent (GD).

All methods were tested over two different scenarios, both of them using public datasets. The first test scenario uses the CICDDoS 2019 dataset and measures the methods over various types of DDoS attacks, where most of the tested methods achieved similarly good results. The second one uses the CICIDS 2018 dataset, providing different kinds of intrusion attacks for the methods' testing. In this scenario, the results are more heterogeneous since the emulated network is composed of more devices, and these attacks tend to be stealthier than flooding ones. GRU achieved the best overall results in both scenarios, presenting the most balanced outcomes regarding the accuracy, precision, recall, and f-measure. We believe GRU fared better than the other tested methods because of its ability to learn long-term dependencies. Since GRU outperformed all other evaluated methods on classifying legitimate flows, we conclude that this characteristic represents a significant advantage of the GRU compared to them.

Furthermore, we measured the number of flows per second each method can analyze and classify since speed is an essential feature for the proposed system. We compared the outcomes with real data collected from a large-scale network and, added to the results obtained through both test scenarios, concluded that GRU is a promising and feasible approach for anomaly detection in real-world SDN environments.

Finally, we proposed a directed mitigation schema, a straightforward approach based on using the individualized flow information provided by the Detection Module. By identifying the attacker IP, the system can generate an individual drop policy against it. Therefore, although this mitigation schema proved to be efficient in the evaluated test scenarios, it is directly influenced by the detection method's performance.

For future work, we intend to use the GRU method as a multi-label classifier, able not only to detect the occurrence of anomalies but also to identify them. Moreover, we want to estimate and evaluate a drop time window usage on the Mitigation Module, calculating the optimal time to minimize the computational cost and improve the mitigation outcomes. We believe that these modifications can significantly improve the performance of the proposed system. Finally, we intend to compare the presented method

with other learning approaches, such as the ensembles algorithms and Deep Reinforcement Learning.

Acknowledgements

This study has been partially supported by the National Council for Scientific and Technological Development (CNPq) of Brazil under Grant of Project 310668/2019-0; 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; and by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) by the granting of a scholarship through the "Programa de Doutorado Sanduíche no Exterior (PDSE) 2019". Finally, this work was supported by Federal University of Paraná (UFPR) under Project Banpesq/2014016797.

References

Abdulhammed, R., Faezipour, M., Abuzneid, A., AbuMallouh, A., 2019. Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic. IEEE Sensors Letters 3, 1–4. doi:10.1109/LSENS.2018.2879990.

Aldweesh, A., Derhab, A., Emam, A.Z., 2020. Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. Knowledge-Based Systems 189, 105124. URL: http://www.sciencedirect.com/science/article/pii/S0950705119304897, doi:https://doi.org/10.1016/j.knosys.2019.105124.

Bengio, Y., Simard, P., Frasconi, P., 1994. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks 5, 157–166. doi:10.1109/72.279181.

Bera, S., Misra, S., Roy, S.K., Obaidat, M.S., 2018. Soft-wsn: Software-defined wsn management system for iot applications. IEEE Systems Journal 12, 2074–2081. doi:10.1109/JSYST.2016.2615761.

Bereziński, P., Jasiul, B., Szpyrka, M., 2015. An entropy-based network anomaly detection method. Entropy 17, 2367–2408. doi:10.3390/e17042367.

Carvalho, L.F., ao, T.A., de Souza Mendes, L., Proença, M.L., 2018. An ecosystem for anomaly detection and mitigation in software-defined networking. Expert Systems with Applications 104, 121 – 133. doi:https://doi.org/10.1016/j.eswa.2018.03.027.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar. pp. 1724–1734. doi:10.3115/v1/D14-1179.

Chowdhury, A., Raut, S.A., Narman, H.S., 2019. Da-drls: Drift adaptive deep reinforcement learning based scheduling for iot resource management. Journal of Network and Computer Applications 138, 51 – 65. doi:https://doi.org/10.1016/j.jnca.2019.04.010.

Correa Chica, J.C., Imbachi, J.C., Botero Vega, J.F., 2020. Security in sdn: A comprehensive survey. Journal of Network and Computer Applications 159, 102595. doi:https://doi.org/10.1016/j.jnca.2020.102595.

Cortez, P., Rio, M., Rocha, M., Sousa, P., 2006. Internet traffic forecasting using neural networks, in: The 2006 IEEE International Joint Conference on Neural Network Proceedings, pp. 2635–2642. doi:10.1109/IJCNN.2006.247142.

da Costa, K.A., Papa, J.P., Lisboa, C.O., Munoz, R., de Albuquerque, V.H.C., 2019. Internet of things: A survey on machine learning-based intrusion detection approaches. Computer Networks 151, 147 – 157. doi:https://doi.org/10.1016/j.comnet.2019.01.023.

Daneshgadeh Çakmakçı, S., Kemmerich, T., Ahmed, T., Baykal, N., 2020. Online ddos attack detection using mahalanobis distance and kernel-based learning algorithm. Journal of Network and Computer Applications 168, 102756. doi:https://doi.org/10.1016/j.jnca.2020.102756.

De Assis, M.V.O., Novaes, M.P., Zerbini, C.B., Carvalho, L.F., Abrão, T., Proença, M.L., 2018. Fast defense system against attacks in software defined networks. IEEE Access 6, 69620–69639. doi:10.1109/ACCESS.2018.2878576.

Divyatmika, Sreekesh, M., 2016. A two-tier network based intrusion detection system architecture using machine learning approach, in: 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), pp. 42–47. doi:10.1109/ICEEOT.2016.7755404.

Farris, I., Taleb, T., Khettab, Y., Song, J., 2019. A survey on emerging sdn and nfv security mechanisms for iot systems. IEEE Communications Surveys Tutorials 21, 812–837. doi:10.1109/COMST.2018.2862350.

Fernandes, Jr., G., Rodrigues, J.J., Carvalho, L.F., Al-Muhtadi, J.F., Proença, Jr., M.L., 2019. A comprehensive survey on network anomaly detection. Telecommun. Syst. 70, 447–489. doi:10.1007/s11235-018-0475-8.

Fukuda, K., Heidemann, J., Qadeer, A., 2017. Detecting malicious activity with dns backscatter over time. IEEE/ACM Transactions on Networking 25, 3203–3218. doi:10.1109/TNET.2017.2724506.

Gkountis, C., Taha, M., Lloret, J., Kambourakis, G., 2017. Lightweight algorithm for protecting sdn controller against ddos attacks, in: 2017 10th IFIP Wireless and Mobile Networking Conference (WMNC), pp. 1–6. doi:10.1109/WMNC.2017.8248858.

Guo, Y., Ji, T., Wang, Q., Yu, L., Min, G., Li, P., 2020. Unsupervised anomaly detection in iot systems for smart cities. IEEE Transactions on Network Science and Engineering , 1–1doi:10.1109/TNSE.2020.3027543.

Hajiheidari, S., Wakil, K., Badri, M., Navimipour, N.J., 2019. Intrusion detection systems in the internet of things: A comprehensive investigation. Computer Networks 160, 165 – 191. doi:https://doi.org/10.1016/j.comnet.2019.05.014.

He, T., Droppo, J., 2016. Exploiting lstm structure in deep neural networks for speech recognition, in: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5445–5449. doi:10.1109/ICASSP.2016.7472718.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780. doi:10.1162/neco.1997.9.8.1735.

Jozefowicz, R., Zaremba, W., Sutskever, I., 2015. An empirical exploration of recurrent network architectures, in: Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, JMLR.org. p. 2342–2350.

Kao, J., Jiang, J., 2019. Anomaly detection for univariate time series with statistics and deep learning, in: 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), pp. 404–407. doi:10.1109/ECICE47484.2019.8942727.

Khan, I.A., Pi, D., Yue, P., Li, B., Khan, Z.U., Hussain, Y., Nawaz, A., 2020. Efficient behaviour specification and bidirectional gated recurrent units-based intrusion detection method for industrial control systems. Electronics Letters 56, 27–30. doi:10.1049/el.2019.3008.

Kornycky, J., Abdul-Hameed, O., Kondoz, A., Barber, B.C., 2017. Radio frequency traffic classification over wlan. IEEE/ACM Transactions on Networking 25, 56–68. doi:10.1109/TNET.2016.2562259.

Kwon, D., Natarajan, K., Suh, S.C., Kim, H., Kim, J., 2018. An empirical study on network anomaly detection using convolutional neural networks, in: 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), pp. 1595–1598. doi:10.1109/ICDCS.2018.00178.

Lei, Y., 2017. Network anomaly traffic detection algorithm based on svm, in: 2017 International Conference on Robots Intelligent System (ICRIS), pp. 217–220. doi:10.1109/ICRIS.2017.61.

Liu, S., Chen, X., Peng, X., Xiao, R., 2019. Network log anomaly detection based on gru and svdd, in: 2019 IEEE Intl Conf on Parallel Distributed Processing with Applications, Big Data Cloud Computing, Sustainable Computing Communications, Social Computing Networking (ISPA/BDCloud/SocialCom/SustainCom), pp. 1244–1249. doi:10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00177.

Lopez-Martin, M., Carro, B., Lloret, J., Egea, S., Sanchez-Esguevillas, A., 2018. Deep learning model for multimedia quality of experience prediction based on network flow packets. IEEE Communications Magazine 56, 110–117. doi:10.1109/MCOM.2018.1701156.

Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., Lloret, J., 2017. Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in iot. Sensors 17. doi:10.3390/s17091967.

Maenhaut, P., Moens, H., Volckaert, B., Ongenae, V., Turck, F.D., 2017. Resource allocation in the cloud: From simulation to experimental validation, in: 2017 IEEE 10th International Conference on Cloud Computing (CLOUD), pp. 701–704. doi:10.1109/CLOUD.2017.96.

Maziku, H., Shetty, S., Nicol, D.M., 2019. Security risk assessment for sdn-enabled smart grids. Computer Communications 133, 1 – 11. doi:https://doi.org/10.1016/j.comcom.2018.10.007.

McDermott, C.D., Majdani, F., Petrovski, A.V., 2018. Botnet detection in the internet of things using deep learning approaches, in: 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. doi:10.1109/IJCNN.2018.8489489.

Nam, K., Kim, K., 2018. A study on sdn security enhancement using open source ids/ips suricata, in: 2018 International Conference on Information and Communication Technology Convergence (ICTC), pp. 1124–1126. doi:10.1109/ICTC.2018.8539455.

Nanda, S., Zafari, F., DeCusatis, C., Wedaa, E., Yang, B., 2016. Predicting network attack patterns in sdn using machine learning approach, in: 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 167–172. doi:10.1109/NFV-SDN.2016.7919493.

Orion, R.G., 2020. Ip flow data from a large-scale network: feasibility test. URL: `http://www.uel.br/grupos/orion/datasets.html`.

Pena, E.H.M., Barbon, S., Rodrigues, J.J.P.C., Proença, M.L., 2014. Anomaly detection using digital signature of network segment with adaptive arima model and paraconsistent logic, in: 2014 IEEE Symposium on Computers and Communications (ISCC), pp. 1–6. doi:10.1109/ISCC.2014.6912503.

Proença, M.L., Zarpelao, B.B., Mendes, L.S., 2005. 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), pp. 290–295. doi:10.1109/AICT.2005.26.

Qin, G., Chen, Y., Lin, Y., 2018. Anomaly detection using lstm in ip networks, in: 2018 Sixth International Conference on Advanced Cloud and Big Data (CBD), pp. 334–337. doi:10.1109/CBD.2018.00066.

Qu, Z., Su, L., Wang, X., Zheng, S., Song, X., Song, X., 2018. A unsupervised learning method of anomaly detection using gru, in: 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), pp. 685–688. doi:10.1109/BigComp.2018.00126.

Sharafaldin, I., Gharib, A., Habibi Lashkari, A., Ghorbani, A., 2017. Towards a reliable intrusion detection benchmark dataset. Software Networking 2017, 177–200. doi:10.13052/jsn2445-9739.2017.009.

Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A., 2018. Toward generating a new intrusion detection dataset and intrusion traffic characterization, in: Proceedings of the 4th International Conference on Information Systems Security and Privacy - Volume 1: ICISSP,, INSTICC. SciTePress. pp. 108–116. doi:10.5220/0006639801080116.

Sharafaldin, I., Lashkari, A.H., Hakak, S., Ghorbani, A.A., 2019. Developing realistic distributed denial of service (ddos) attack dataset and taxonomy, in: 2019 International Carnahan Conference on Security Technology (ICCST), pp. 1–8. doi:10.1109/CCST.2019.8888419.

Shuying Chang, Xuesong Qiu, Zhipeng Gao, Feng Qi, Ke Liu, 2010. A flow-based anomaly detection method using entropy and multiple traffic features, in: 2010 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT), pp. 223–227. doi:10.1109/ICBNMT.2010.5705084.

Sidki, L., Ben-Shimol, Y., Sadovski, A., 2016. Fault tolerant mechanisms for sdn controllers, in: 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 173–178. doi:10.1109/NFV-SDN.2016.7919494.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15, 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

Sun, D., Fu, M., Zhu, L., Li, G., Lu, Q., 2016. Non-intrusive anomaly detection with streaming performance metrics and logs for devops in public clouds: A case study in aws. IEEE Transactions on Emerging Topics in Computing 4, 278–289. doi:10.1109/TETC.2016.2520883.

Tatang, D., Quinkert, F., Frank, J., Röpke, C., Holz, T., 2017. Sdn-guard: Protecting sdn controllers against sdn rootkits, in: 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 297–302. doi:10.1109/NFV-SDN.2017.8169856.

Theodorou, T., Mamatas, L., 2017. Coral-sdn: A software-defined networking solution for the internet of things, in: 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 1–2. doi:10.1109/NFV-SDN.2017.8169870.

Wang, P., Yang, L.T., Nie, X., Ren, Z., Li, J., Kuang, L., 2020. Data-driven software defined network attack detection : State-of-the-art and perspectives. Information Sciences 513, 65 – 83. doi:https://doi.org/10.1016/j.ins.2019.08.047.

Wijnhoven, R.G.J., de With, P.H.N., 2010. Fast training of object detection using stochastic gradient descent, in: 2010 20th International Conference on Pattern Recognition, pp. 424–427. doi:10.1109/ICPR.2010.112.

Xie, X., Wang, B., Wan, T., Tang, W., 2020. Multivariate abnormal detection for industrial control systems using 1d cnn and gru. IEEE Access 8, 88348–88359. doi:10.1109/ACCESS.2020.2993335.

Xu, J., Wang, L., Xu, Z., 2020. An enhanced saturation attack and its mitigation mechanism in software-defined networking. Computer Networks 169, 107092. doi:https://doi.org/10.1016/j.comnet.2019.107092.

Yadav, S., Selvakumar, S., 2015. Detection of application layer ddos attack by modeling user behavior using logistic regression, in: 2015 4th International Conference on Reliability, Infocom Technologies and

Optimization (ICRITO) (Trends and Future Directions), pp. 1–6. doi:10.1109/ICRITO.2015.7359289.

Yoon, S., Kim, J., 2017. Remote security management server for iot devices, in: 2017 International Conference on Information and Communication Technology Convergence (ICTC), pp. 1162–1164. doi:10.1109/ICTC.2017.8190885.

Zehra, U., Shah, M.A., 2017. A survey on resource allocation in software defined networks (sdn), in: 2017 23rd International Conference on Automation and Computing (ICAC), pp. 1–6. doi:10.23919/IConAC.2017.8082092.

Zhang, S., Wang, Y., Zhou, W., 2019. Towards secure 5g networks: A survey. Computer Networks 162, 106871. doi:https://doi.org/10.1016/j.comnet.2019.106871.

Zhang, X., Xie, L., Yao, W., 2020. Spatio-temporal heterogeneous bandwidth allocation mechanism against ddos attack. Journal of Network and Computer Applications 162, 102658. doi:https://doi.org/10.1016/j.jnca.2020.102658.

Zhang, X., Zhang, Y., Zhang, L., Wang, H., Tang, J., 2018. Ballistocardiogram based person identification and authentication using recurrent neural networks, in: 2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), pp. 1–5. doi:10.1109/CISP-BMEI.2018.8633102.