Article

Hybrid Deep Learning Approach for Automatic DoS/DDoS Attacks Detection in Software-Defined Networks

Hani Elubeyd * and Derya Yiltas-Kaplan

Department of Computer Engineering, Istanbul University-Cerrahpaşa, Istanbul 34320, Turkey
* Correspondence: hani.elubeyd@gmail.com

Abstract: This paper proposes a hybrid deep learning algorithm for detecting and defending against DoS/DDoS attacks in software-defined networks (SDNs). SDNs are becoming increasingly popular due to their centralized control and flexibility, but this also makes them a target for cyberattacks. Detecting DoS/DDoS attacks in SDNs is a challenging task due to the complex nature of the network traffic. To address this problem, we developed a hybrid deep learning approach that combines three types of deep learning algorithms. Our approach achieved high accuracy rates of 99.81% and 99.88% on two different datasets, as demonstrated through both reference-based analysis and practical experiments. Our work provides a significant contribution to the field of network security, particularly in the area of SDN. The proposed algorithm has the potential to enhance the security of SDNs and prevent DoS/DDoS attacks. This is important because SDNs are becoming increasingly important in today’s network infrastructure, and protecting them from attacks is crucial to maintaining the integrity and availability of network resources. Overall, our study demonstrates the effectiveness of a hybrid deep learning approach for detecting DoS/DDoS attacks in SDNs and provides a promising direction for future research in this area.

Keywords: hybrid deep learning; DoS/DDoS attacks; software-defined networks (SDNs); cyberattacks

1. Introduction

A precondition to examining DDoS attacks in software-defined networks (SDNs) is establishing a DDoS attack definition. The foundational level classified a denial of service (DoS) attack as an attack against a network structure that causes a server to be disabled from servicing its clients [1]. The researchers argue that DoS attacks can send a high volume of requests to a server to slow it down, with sizeable invalid data packets, or send spoofed or invalid IP address requests to ensure the flood server. Ref. [2] supported Tang and [1] by revealing that the main objective of launching a DoS attack is to disrupt the availability of network resources for legitimate users through one of several possible strategies. For instance, attackers send messages to exploit vulnerabilities leading to paralysis of the network systems, or attackers send a high volume of regular messages to a single node that exhausts the system resources and, as such, crashes the entire system [2]. The most recent DDoS attack was against Amazon Web Services (AWS) in 2020, but that attack was mitigated, although it involved up to 2.3 Tbps of data [3]. Another study [4] supported [3] by revealing that the DDoS attack involved hijacked Connectionless Lightweight Directory Access Protocol (CLDAP) servers, which amplified the attack by up to 70 times its size.

In recent years, software-defined networking (SDN) has emerged as a promising approach for improving network efficiency and flexibility. However, SDN-based networks have also been subject to DDoS attacks, which can disrupt the delivery of essential services to customers. DDoS attacks involve overwhelming a network with traffic from multiple sources, rendering it unavailable to legitimate users. This has serious consequences for organizations and businesses that rely on networked systems for their operations and highlights the need to design and implement effective security measures to detect the risk of DDoS attacks on SDN networks.
Previous research has explored various approaches to addressing this problem, including non-deep learning solutions such as packet entropy techniques [5] and time-based methods [6]. While these approaches have successfully mitigated DDoS attacks, there is still room for improvement in accuracy and efficiency. In particular, the adoption of deep learning algorithms has the potential to significantly improve the effectiveness of security measures for SDN networks. For example, the GA [7] and RNN [8] algorithms have been implemented as unique deep learning approaches to addressing DDoS attacks in SDN networks. This was a significant contribution, as low-rate DDoS attacks can be particularly difficult to detect due to their low volume and slow rate of traffic. By considering both types of attacks, our approach was able to provide a more comprehensive and robust analysis of the security challenges facing SDN networks. One of the key differences between our approach and other existing approaches was that while they may have focused mainly on detecting volumetric DDoS attacks, our approach considered both volumetric and low-rate DDoS attacks in the context of SDN networks.

Additionally, our approach employed a hybrid deep learning approach that combined multiple algorithms and datasets in order to improve the accuracy and efficiency of our analysis. This was in contrast to other approaches that may have relied on a single algorithm or dataset, which may be less robust and less able to handle the complexity and variability of real-world data. By using a hybrid approach, we aimed to provide a more reliable and effective solution for detecting and defending against DDoS attacks in SDN networks.

This research makes a significant scientific contribution to the field of DDoS attack detection and defense in SDN networks by proposing a hybrid deep learning approach that combines multiple algorithms and datasets to improve accuracy and efficiency. This approach is novel in its ability to detect both volumetric and low-rate DDoS attacks and to handle the complexity and variability of real-world data. The results of this study demonstrate the effectiveness of deep learning techniques in addressing the security challenges facing SDN networks and highlight the potential for further improvements in this area. Overall, this research contributes to the development of more reliable and effective security measures for SDN networks, which are critical for ensuring the availability and integrity of networked systems in today’s interconnected world.

The structure of the remainder of this paper is as follows: In the Section 2, we review relevant literature and previous work on the topic of DDoS attacks and security measures for SDN networks. The Section 3 describes the research method and approach used in this study, including the specific deep learning algorithms and datasets that were employed. In the Section 4, we present the results of our analysis and comparison of different security measures for SDN networks. Finally, in the Section 5, we provide a summary of our findings and conclusions.

2. Related Work

In the field of securing software-defined networks (SDNs) against DDoS attacks, researchers have employed various deep learning techniques. Some studies have utilized single algorithms, such as convolutional neural networks (CNNs) or long short-term memory (LSTM) approaches, while others have employed hybrid algorithms combining multiple techniques. According to [9,10], hybrid deep learning algorithms are the most widely used approaches for detecting and mitigating attack traffic within SDNs. In this section, we will discuss different approaches to demonstrate the significance of deep learning in the context of securing SDNs against DDoS attacks.

2.1. Individual Deep Learning Algorithms Implementation

Other researchers have focused on comparing how different deep learning algorithms performed in DDoS attack detection in SDNs. A case in point was [11]. They reached the simple neural network, CNN, and RNN in DDoS attack detection in the CSE-CIC-IDS2018 dataset, which simulated diverse attack scenarios, including brute-force, DoS, infiltration, DDoS, Heartbleed, and botnet attacks. Findings from the study revealed that simple neural
networks outperformed CNN and RNN, attributed to the detection of malware by generating an accuracy of 82% and a precision of 42%. The authors argued that the more popular RNN and CNN could have performed better due to the challenges of overfitting, which led to high false positives and negative rates [11]. In another study, Ref. [12] also compared the performance of several algorithms to facilitate the detection of DDoS attacks in SDN environments. In particular, it was also considering the Support Vector Machine (SVM), Naïve Bayes (NB), Artificial Neural Network (ANN), and K-Nearest Neighbor (KNN) classification models. In addition, the researchers adopted feature selection techniques to simplify the models used in the experiments to ensure model simplification, enhance their interpretation, and facilitate a shorter training time [12]. Findings reported showed that wrapper feature selection combined with the KNN classifier generated the highest score of 98.3% accuracy in DDoS-attack detection. Analytically, the results underscored the performance improvement associated with combining deep learning with feature selection techniques when detecting DDoS attacks to facilitate reduced load processing times [12]. Furthermore, there has been critical research [13]; the authors proposed an approach against DDoS attacks by adopting the Gated Recurrent Units (GRU) method and compared it with other deep learning algorithms by explaining how their approach outperforms. Findings from the study showed that the GRU-RNN did not adversely affect network performance and led to high accuracy performances of 99% and 89% for the CICIDS2017 and NSL-KDD datasets (Tang et al., 2019). In previous research, Ref. [14] reported an accuracy of 99.63% for the CNN-RF model and 99.58% for the RF-MLP using the same dataset—CICIDS2017. Analytically, the comparison of [1,14] revealed, using the same dataset, different hybrid deep learning algorithms that generated similar performances—99% for both the CNN-RF and GRU-RNN in both cases. Such insights are essential in the current study, as they reveal that differences in implementation of the hybrid deep learning algorithms did not vary significantly despite using different deep learning algorithms.

2.2. Hybrid Deep Learning Techniques

This section discusses the hybridization techniques and deep learning algorithms to ensure the security of SDNs against DDoS attacks.

Efficient Hybridization Technique for Intrusion Detection Systems has become a critical technology to safeguard against malicious threats in cyberspace. Many soft computing approaches have been employed to enhance the effectiveness of Intrusion Detection Systems (IDS). However, the high dimensionality of network traffic data, dynamic attack patterns, and the need for multiple classifiers to detect various forms of attacks remain significant challenges [15–17]. To address these challenges, this paper proposes a hybridization technique that combines supervised and unsupervised learning techniques. K-means clustering is used to classify the data into normal and attack classes, and wrapper feature selection with a genetic algorithm is employed to address the high dimensionality of the data. The input data are then classified with a support vector machine (SVM) [15]. Another proposed technique for feature selection involves the use of the metaheuristic Bat algorithm and PCA [16]. Lastly, a combination of blockchain technology and machine learning techniques was used to manage datasets and detect network communications for intrusion detection systems [17]. The proposed techniques were shown to achieve high accuracy and low false alarm rate, with promising benefits of robustness, low computational cost, and generalization by reducing possible overfitting.

In terms of hybrid deep learning, two main ideas addressed include the integration of more than one deep learning algorithm to detect attack traffic and the combination of the algorithms with other network security solutions such as IDS and IPS devices. To begin with, Ref. [9] argued that deep learning techniques are essential in detecting DDoS attacks within SDNs to identify the attacks as anomalies within legitimate traffic. Therefore, in discussing the importance of deep learning in securing SDNs, there is a need to view the attacks as anomalies that result in traffic generation. The study by [9]
proposed a deep learning approach that regarded the implementation of Convolution Neural Network (CNN) and Long Short-Term Memory (LSTM) algorithms to detect slow DDoS attacks in SDNs. In the research, datasets to emulate the slow DDoS attack traffic flow were generated and leveraged the ability of SDN switches to detect traffic flow statistics. The training was performed for the CNN-LSTM model and was validated while also undertaking hyperparameter tuning. The performance of the CNN-LSTM was compared against a MultiLayer Perceptron (MLP) and 1-Class Support Vector Machine (1-SVM). The generated results revealed that the CNN-LSTM outperformed the MLP and 1-SVM in terms of accuracy, precision, recall, and specificity [9].

Another study [14] supported [10] by revealing that combining several deep learning algorithms such as RF, CNN, and MLP methods improved DDoS attack detection in IoT networks and devices. In the research, Ref. [14] reported a higher accuracy of 99.63% for the hybrid model combining the Random Forests and Convolutional Neural Networks compared to 99.58% accuracy obtained from the combination of Random Forests and the Multilayer Perceptron. Such insights underscored the value of CNN-RF hybrid models in outperforming other variants in DDoS detection in SDN networks.

Other researchers, such as [18], combined the CNN deep learning algorithm with information entropy to detect DDoS attacks in SDNs to distinguish between legitimate and attack traffic. The outcomes from the research indicated that the hybrid model generated high performance in detecting traffic anomalies. In the study, Ref. [18] argued that combining CNN and information entropy was essential to leverage their advantages in DDoS attack detection. The low-complexity advantages of information entropy were combined with the high accuracy of the CNN algorithms, thereby facilitating DDoS attack detection in the SDN controller and guaranteeing the security of the SDN network. Ref. [19] also conducted a similar study to [18], whereby information entropy was combined with a CNN algorithm to detect DDoS attacks. In their study, Ref. [19] used information entropy to see suspicious ports and components in coarse granularity, whereas CNN was adopted to distinguish legitimate and attack traffic. Findings from the research revealed high accuracy in detecting the anomaly traffic at 98.98%, which underscored the robustness of combining information entropy and CNN techniques in detecting attack traffic. Therefore, a similarity between [18,19] emerged from the fact that both studies advocated combining information entropy with CNNs to detect attack traffic within SDNs. As a result, there was better performance and accuracy regarding mitigating DDoS attacks. Further analysis, however, indicated that the shortcoming of reliance on deep learning algorithms on their own arose from high training costs and low efficiency despite their high accuracy. Analytically, such findings indicate a need to identify more alternative solutions to reduce the disadvantages of deep learning algorithms.

Ref. [20] reiterated [21] by demonstrating that integrating a deep learning algorithm enhanced IDS systems. Ref. [22] further demonstrated the effectiveness of the Stacked Autoencoders (SAE) deep learning algorithm combined with a Snort IDS in optimizing the detection accuracy of DDoS attack detection within SDN environments. By implementing the hybrid algorithm, the study observed a high accuracy of 95%.

Additionally, hybrid deep learning algorithms have shown promising results in detecting low-rate DDoS attacks in software-defined networks (SDNs). These attacks are characterized by a low volume of traffic but a high frequency, making them difficult to detect using traditional approaches such as threshold-based systems [3]. Another hybrid deep learning method is the integration of a CNN with a deep belief network (DBN), as presented by [23]. This approach achieved a detection rate of 96.7% and a false positive rate of 1.2% on a simulated SDN dataset.

Finally, the analysis also indicated that deep learning algorithms could better detect DDoS attacks in SDN environments [6–18]. The research emerged that deep learning techniques outperformed classical machine learning algorithms, as they did not involve human interaction to improve their performance but instead relied on artificial neural networks [24]. The evaluation also demonstrated that hybrid deep learning techniques
outperformed single algorithms even in scenarios where the models comprised deep learning algorithms. Therefore, in this article, the superior performance of our hybrid approach in detecting DDoS attacks in SDN environments is examined further to protect SDN environments. In particular, the research focused on developing a model comprising deep learning algorithms to ensure high performance in detecting DDoS attacks within SDN environments.

3. The Proposed Method

It is essential to highlight that investigating different IP flow features (or dimensions) enriches traffic analysis by providing relevant information about the communication. Several different approaches use this characteristic to improve the performance of anomaly detection. Still, most of them use a set of manually selected features, 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 deep learning approaches that can analyze several flow features and automatically give more importance to those dimensions that most impact the classification outcomes.

3.1. Dataset Types

In SDN, various types of datasets are used to test and evaluate the effectiveness of DDoS attack detection and prevention mechanisms. Here are some examples of datasets commonly used for this purpose.

3.1.1. Real-World Datasets

These datasets are collected from actual SDN networks and contain real traffic patterns and attack behaviors. An example of this dataset is the CIC-DDoS2019 dataset, which was collected from a large enterprise network and includes benign and DDoS attack traffic. This dataset has been widely used in research on DDoS attack detection and prevention in SDN networks (e.g., [25]).

3.1.2. Synthetic Dataset

Synthetic data are generated algorithmically from actual data by statistical properties. However, the synthetic dataset for DDoS attacks in SDN, which injects attack activity into the dataset, depends on tunable attributes such as the number of nodes attacked [26]. Moreover, the synthetic dataset should include accurate values to guarantee privacy and anonymity to avoid the fact that synthetic traffic causes a normality model, which may lead to incorrect results [27]. In this section also, we mention some articles that used more than one dataset source as the following:

- The work of [29] proposed a multilayer perceptron (MLP) neural network to detect HTTP-based slow-rate attacks. For their proposed method, they used three different datasets: CTU dataset, WIDE dataset, and generated MANET dataset.

Several approaches utilize public tools, such as Scapy and Hping3, to generate synthetic datasets for testing DDoS attack detection and prevention mechanisms in software-defined networking (SDN). These approaches are often fast in their calculation and are simple to analyze. However, there are some limitations to consider. For example, the datasets generated by these approaches may be small in size, potentially leading to insufficient or inaccurate results. Additionally, the number of features extracted may not be sufficient to cover all attack behavior, which could impact the overall effectiveness of the approach.

3.1.3. Dataset Selection

Thus far, the ML/DL model highly depends on feature selection and the quality and quantity of training data. Collecting data from real-world scenarios is a good idea when starting with a real-world dataset. Nevertheless, because of the investigation, it was found...
that all the obtained records were from the IDs. In short, these data are not exclusive to our case study. Second, we focus on SDN networks, but most datasets are generated from traditional networks.

For this reason, we can choose to generate a dataset dedicated to DDoS traffic and SDN networks. To generate the dataset, the Ryu controller must be configured as a traffic monitor to generate regular and DDoS traffic. We can then collect and save the information in a “dataset” file. Moreover, we adopted CICDDoS2019, an authorized set of features introduced by the Canadian Institute for Cybersecurity.

3.2. Tools and Environment

For this section, we mentioned tools and the environment as the following:

1—Hypervisor: Our choice fell on VirtualBox to simulate these nodes and create a virtual network close to a real network.

2—UBUNTU Server: Installed a controller and switch on the ubuntu server in virtual machines.

3—Controller: Feature richness, ease of use, and implementation language play a significant role in controller choice. We chose the RYU SDN controller for the following reasons:

- Compatibility with UNIX systems also supports Python language;
- Simple to install and deploy due to availability in Python environments such as Anaconda, which provides the required dependencies, and ease of installation using “pip” as package manager;
- Support all versions of OpenFlow;
- Ability to monitor traffic.

4—Switch: Open vSwitch is well suited to operate as a virtual switch in VM environments and supports several Linux-based virtualization technologies, including VirtualBox.

5—SDN environment: Mininet provides SDN’s virtual examination and development environment. It enables SDN development on any laptop or PC. It provides an extensible Python API for creating networks and experimenting. The similarity of the programming language between the Ryu controller and the Mininet environment makes deploying a virtual testbed more flexible. We used the Ryu controller rather than the default Mininet controller and the built-in Open vSwitch switch.

All environmental elements above were employed to generate our dataset and check our proposed approach’s efficiency.

Moreover, after uploading our two datasets on Google Drive, we linked them with Google Colab to check our results and to evaluate our algorithm’s performance.

6—Google Colab and Google Drive: To execute our Python code and follow the organized stages of our hybrid approach, from data pre-processing to presenting the results, we used Google Colab. Additionally, we utilized Google Drive to host our datasets and linked them with Google Colab to ensure a consistent environment for comparison with other research. This allowed us to efficiently run our code and easily access and manipulate our datasets throughout the project.

7—CICDDoS2019 dataset: The Canadian Institute for Cybersecurity created a dataset comprising thousands of synthesized DDoS attack scenarios based on actual attacks documented in 2019, with a particular focus on volumetric attacks that aim to overwhelm a network or server with a high volume of traffic, causing denial of service to legitimate users. By using this dataset, we were able to evaluate the performance of our proposed hybrid deep learning approach in detecting volumetric DDoS attacks in software-defined networking (SDN) environments. Figure 1 Illustrates the percentage breakdown of traffic types in the dataset.
The CICDDoS2019 dataset is a recent addition to the field of cybersecurity, which includes a large-scale, labeled dataset of DDoS attacks on IoT devices. The dataset contains 15 different types of network-based attacks, along with normal traffic, and comprises a total of 2,000,753 samples. The data were collected in a lab environment, emulating a real-world network. The features of the dataset include network flow features, transport layer features, and application layer features, and it has a total of 115 features. One of the unique features of this dataset is its focus on DDoS attacks against IoT devices, which are becoming increasingly common and challenging to detect. The inclusion of various types of attacks and normal traffic provides a comprehensive representation of network traffic, making it suitable for developing and testing intrusion detection systems. Moreover, the large number of samples and variety of attacks in the dataset enables the development of robust and accurate machine learning models for detecting attacks in IoT networks. In summary, the CICDDoS2019 dataset provides a valuable resource for cybersecurity researchers and practitioners to develop and evaluate intrusion detection systems for IoT networks. The dataset’s features, samples, and variety of attacks make it suitable for addressing the challenges of detecting DDoS attacks against IoT devices in real-world environments.

The dataset was pre-processed by removing null and infinite values, unchanged features, and features that were deemed useless or highly correlated with other features. After pre-processing, the remaining features included Source Port, Destination Port, Protocol, Flow Duration, Total Fwd Packets, Total Backward Packets, Total Length of Fwd Packets, Fwd Packet Length Max, Fwd Packet Length Min, Fwd Packet Length Std, and so on. The final set of features also included Down/Up Ratio, Init_Win_bytes_forward, Init_Win_bytes_backward, Active Mean, Active Std, Active Max, Active Min, Idle Std, Inbound, and Label. These features were used for the classification of network traffic in the CICDDoS2019 dataset.

Overall, the use of this dataset allowed us to specifically focus on the challenges of detecting volumetric DDoS attacks in SDN environments, providing valuable insights for future research in this area.

8—Generated dataset: In this work, we used a simulation approach to build a training dataset specifically designed to capture low-rate DDoS attacks in software-defined networking (SDN) environments. Many publicly available datasets are unrealistic and lack diversity in attack types, making it difficult to effectively train deep learning models on real-world scenarios. To address this issue, we generated traffic over a two-day period of normal/benign traffic and 25 min of DDoS traffic, divided into three attack periods of 8 min each. We labeled benign and malicious packets during the benign and malicious
traffic generation to simplify the deep learning process. To mimic an SDN environment, we collected the dataset by adding simulated traffic flow entries from the switch flow table. The data were then pre-processed and converted into a .csv file format that could be read by the Python-based deep learning module. Figure 2 shows the percentage breakdown of benign and attack traffic in the dataset.

**Figure 2.** The percentage of traffic types of generated dataset.

This approach allowed us to create a realistic and diverse dataset for training a deep learning model to detect low-rate DDoS attacks in SDN environments. Additionally, Figure 3 demonstrates the stages of the generation of the dataset.

**Figure 3.** Stages of generating the dataset.
Overall, this approach allowed us to create a realistic and diverse dataset that accurately reflects the challenges of detecting low-rate DDoS attacks in SDN environments, providing a strong foundation for the development of effective deep learning models for this task.

3.3. Generation Traffic Tools

To generate the training dataset comprising both normal and DDoS flows, we utilized a variety of tools. The Ping tool was used to test the reachability of a host on an IP network, while the iPerf tool served as an active measurement mechanism for determining the maximum possible bandwidth on IP networks. The SimpleHTTPServer Python Module was employed to create a primary web server with minimal effort, and the Hping3 tool allowed us to generate various types of attacks. These tools enabled us to effectively simulate a range of normal and DDoS traffic scenarios, providing a diverse and realistic dataset for training a deep learning model to detect low-rate DDoS attacks in software-defined networking (SDN) environments.

3.4. Proposed Hybrid Deep Learning Approach

In our research, we propose a hybrid deep learning model for detecting distributed denial of service (DDoS) attacks in software-defined networking (SDN). The choice of deep learning algorithms depends on the specific characteristics of the problem being addressed. In the case of detecting DDoS attacks in SDN networks, a combination of a 1D convolutional neural network (CNN), a gated recurrent unit (GRU), and a dense neural network (DNN) was selected because they each have strengths that complement each other in handling the complexities of the problem.

The 1D CNN is a powerful tool for detecting patterns in sequential data, such as network traffic. It is able to automatically extract relevant features from the input data, such as packet size, frequency, and direction, and capture both short-term and long-term patterns. This is especially useful for detecting low-rate DDoS attacks, which can be difficult to detect using traditional methods.

The GRU is a type of recurrent neural network (RNN) that is well suited for handling sequential data with long-term dependencies. It is able to learn and remember patterns over long periods, which is important for detecting subtle changes in network traffic that may be indicative of an attack.

Finally, the DNN is a versatile algorithm that is able to learn complex mappings between input and output data. It is able to handle a wide range of input data types and can be used to model complex relationships between different input features.

Other deep learning algorithms could also be used, but the combination of 1D CNN, GRU, and DNN was found to be particularly effective in detecting and defending against DDoS attacks in SDN networks. The selection of algorithms should be based on the specific problem being addressed and the characteristics of the data being analyzed.

Our approach was able to effectively detect both low-rate and volumetric DDoS attacks in SDN networks. This was achieved through the use of a combination of deep learning algorithms, including a 1D convolutional neural network (CNN), a gated recurrent unit (GRU), and a dense neural network (DNN). Our experience and results have shown that this combination of algorithms is particularly effective in addressing the unique challenges posed by DDoS attacks in SDN networks. By considering both types of attacks, our approach was able to provide a more comprehensive and accurate analysis of the security challenges facing SDN networks. Additionally, our hybrid deep learning approach, which combined multiple algorithms and datasets, further improved the accuracy and efficiency of our analysis. Overall, our approach represents a significant contribution to the field of DDoS attack detection and defense in SDN networks.
The 1D CNN is responsible for extracting features \( x_i \) from the input network traffic data, which is typically in the form of time series data. The features \( x_i \) are extracted using the following convolutional operation:

\[
x_i = f(W \times x_{i-1} + b)
\]

where \( W \) is the weight matrix, and \( b \) is the bias term.

The GRU is a type of recurrent neural network (RNN) that is able to process sequences of data and capture long-term dependencies, making it well suited for analyzing time series data. The GRU computes hidden states \( h_i \) using the following equation:

\[
h_i = (1 - z_i) \times h_{i-1} + z_i \times c_i
\]

where \( z_i \) and \( c_i \) are gate variables computed using sigmoid and tanh activation functions, respectively.

The DNN is a fully-connected neural network that is used to classify the input data as either normal or attack traffic based on the features extracted by the 1D CNN and the temporal dependencies captured by the GRU. The output \( y_i \) of the DNN is computed using the following equation:

\[
y_i = f(W_{out} \times h_i + b_{out})
\]

where \( W_{out} \) and \( b_{out} \) are the weight matrix and bias term for the output layer.

We designed our hybrid deep learning model in this way because we wanted to take advantage of the strengths of each individual component. The 1D CNN is effective at extracting features from time series data, while the GRU is able to capture long-term dependencies in this data. The DNN is then able to use these features and dependencies to classify the input data as normal or attack traffic. This hybrid approach has the advantage of being able to capture both short-term and long-term patterns in the input data, which is important for accurately detecting DDoS attacks in SDN.

To further improve the performance of our model, we also include a MaxPool layer, which uses a max operation to pool the sets of features, resulting in a reduced number of features. This can help to reduce overfitting and improve the generalization performance of the model. Additionally, we incorporate a Leaky layer, which allows for a slight gradient when a unit is inactive. This can help to prevent the model from becoming stuck in a local minimum during training and improve its performance in real-world scenarios.

In addition to the benefits described above, our proposed hybrid deep learning model for detecting distributed DDoS attacks in SDN is also effective in detecting low-rate DDoS attacks and defending against DDoS attacks in general. Low-rate DDoS attacks are a type of attack that involve a small number of compromised devices, sending a relatively low volume of traffic to a target network or server. These attacks can be difficult to detect using traditional machine learning algorithms or other deep learning approaches, as they often do not exhibit the same patterns or characteristics as more traditional DDoS attacks. However, our hybrid deep learning model is able to effectively detect low-rate DDoS attacks by taking into account both short-term and long-term patterns in the input data. This is possible due to the combination of a 1D convolutional neural network (CNN), a gated recurrent unit (GRU), and a dense neural network (DNN) in our model, which allows it to extract and analyze both local and global features in the input data.

Overall, our hybrid deep learning model is able to effectively detect and defend against DDoS attacks in SDN, providing a robust and reliable solution for improving the security and reliability of these networks. By accurately identifying and mitigating DDoS attacks, our model can help to ensure that SDN networks are able to operate effectively and efficiently, even in the face of increasingly sophisticated and targeted attacks.

Figure 4 further illustrates a diagram of our suggested hybrid algorithm architecture, which is composed of the following:
Figure 4. Hybrid model through a high-level and traditional neural network representation.

Figure 5 illustrates a capture from the Google Colab platform, demonstrating the implementation of our proposed model.

4. Observation and Results

This work further identified the best-performing approaches and features the hybrid solutions to detect DDoS attacks in SDN. After running our proposed approach, we considered our results and compared our hybrid algorithm performance and GRU adopted by paper [13]. One reason to compare the performance of our approach with the gated recurrent unit (GRU) approach is that the GRU approach has already been shown to outperform state-of-the-art approaches in this context. Figure 6 illustrates a GRU approach diagram as the authors proposed it.
We adopted the confusion matrix to compute our model performance via accuracy, precision, recall, F1 score, and receiver operating characteristic (ROC) via the following:

4.1. Confusion Matrix

The confusion matrix (shown in Figure 6) is a tool for evaluating the performance of classification models that provides a summary of their binary outputs. It is calculated based on four parameters: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). TP refers to correctly predicted positive values, TN refers to correctly predicted negative values, FP refers to incorrectly predicted positive values, and FN refers to incorrectly predicted negative values. The confusion matrix and its four parameters are illustrated in Figure 7 below.

![Confusion Matrix Visualization](image)

Figure 7. Confusion matrix visualization.

There are several metrics that can be used to evaluate the performance of classification models and summarize their results. These include:

**Accuracy:** This metric measures the proportion of correctly classified instances out of the total number of records. It is calculated as:

$$\text{Accuracy} = \frac{TN + TP}{TN + FP + TP + FN}$$
Precision: Precision is a measure of the model’s performance, with a high precision value indicating better performance. It is calculated as:

\[
\text{Precision} = \frac{TP}{TP + FP}
\]

Recall: Recall, also known as sensitivity or true positive rate, is a measure of the proportion of actual positive instances that are correctly identified by the model. A high recall value, ideally close to 1, is desired for a successful proposed method. It is calculated as:

\[
\text{Recall} = \frac{TP}{TP + FN}
\]

F1 Score: This is a metric that combines precision and recall, with a higher value indicating better performance. It is calculated as:

\[
F1 \text{ Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]

In the context of detecting DDoS attacks in SDN environments, high values for accuracy, precision, recall, and F1 Score are desirable, as they indicate that the model is able to accurately identify and mitigate attack traffic while minimizing false alarms and correctly identifying normal traffic.

4.2. ROC Curve

To evaluate the performance of our proposed model, we used the receiver operating characteristic (ROC) curve, which is a widely accepted evaluation metric for AI models. The ROC curve plots the true positive rate against the false positive rate, and the area under the ROC (AUROC) is a measure of the model’s overall performance. The higher the AUROC, the better the model’s performance compared to other methods. In the context of detecting DDoS attacks in SDN environments, the ROC curve and AUROC are useful metrics for assessing the effectiveness of the model in identifying and mitigating attack traffic while minimizing false alarms and correctly identifying normal traffic. Overall, the ROC curve and AUROC provide valuable insights into the performance of our proposed model and can be used to compare its performance with other approaches.

Loss: In deep learning and machine learning, an inaccurate prediction is known as a loss. The magnitude of the loss indicates how accurate or inaccurate the model’s prediction is. If the loss is high, the model is not providing satisfactory results on the test data; on the other hand, if the loss is low, the model is providing satisfactory results on the test data. In this work, we used Colab and Google Drive to compare the performance of our proposed model on two different datasets that underwent the same pre-processing steps. By using these tools, we were able to effectively evaluate the accuracy and loss of the model and determine its effectiveness in detecting DDoS attacks in SDN networks.

4.3. CICDDoS019 Dataset

Our proposed hybrid algorithm was compared with the GRU algorithm using the evaluation metrics shown in Figure 8. The results indicate that the hybrid algorithm outperforms the GRU algorithm in all metrics, with a particularly significant difference in loss. The accuracy of the hybrid algorithm was 99.81%, while the GRU algorithm achieved an accuracy of 98.86%.
Figure 8. The results of our proposed hybrid method with the GRU method regarding the CICD-DoS2019 dataset.

These results demonstrate the superiority of the hybrid algorithm in detecting DDoS attacks within SDN environments, with a notable improvement in performance compared to the GRU algorithm.

As depicted in Figure 9, our proposed hybrid approach demonstrates superior performance on unseen data, also known as the test data, compared to the GRU method. This suggests that our hybrid approach is more effective at generalizing new data and may be a better choice for real-world applications.
Figure 9. The result or hybrid method with the GRU method regarding the CICDDoS2019 dataset.

The standard procedure for running this algorithm involves training for a set number of epochs, which in this case was 100 epochs as stated in the reference paper. To ensure an optimal training process, we implemented the Early Stopping technique, which monitors the performance of the algorithm on the validation set and stops training if there are no improvements in the data. However, our algorithm achieved the best result at an early stage, as shown in Figure 10, requiring only 10 epochs to reach its maximum performance.

This exceptional efficiency is not uncommon in machine learning and deep learning algorithms, as we have observed in previous studies. Figure 10 illustrates the effectiveness of our algorithm in accomplishing the task with minimal computational resources, as evidenced by the red arrow which indicates the number of epochs employed in the experiment.
Figure 10. The epochs number of our proposed method.

Figure 11 illustrates the number of epochs required for the GRU method to reach its maximum performance, which was found to be 17 epochs. The red arrow in the figure denotes the number of epochs employed in the experiment, serving as a visual cue for this key parameter. The number of epochs can significantly impact the performance of an algorithm, as it determines the amount of training that the model undergoes. In this case, the GRU method required a relatively high number of epochs to reach its optimal performance, as shown in Figure 11.

Figure 11. The number of epochs of the GRU method.

Figure 12 presents the receiver operating characteristic (ROC) curve for the CICDDoS2019 dataset, comparing the performance of our proposed method with the GRU method. Both approaches achieved good results, with our proposed method slightly outperforming the GRU method. Specifically, the area under the curve (AUC) for our proposed method was 0.9999, which is extremely close to the ideal value of 1, while the GRU method achieved an AUC of 0.9973. These results suggest that both methods are effective at distinguishing between normal and attack traffic, with our proposed method exhibiting a slight edge in terms of overall performance.
The values, as shown in Table 1, of this confusion matrix represent the performance of a classification model. It consists of two classes, false and true, and the model’s predictions were compared against the true labels. The matrix shows that there were a total of 6219 instances, out of which 3090 were correctly classified as false and 3129 were correctly classified as true. This gives an overall accuracy of 1.00 or 100%. The macro average and weighted average for precision, recall, and F1 score is also 1.00, indicating that the model performed equally well for both classes. The high accuracy and precision values suggest that the model is performing very well, and there are no false positives or false negatives.

### Table 1. Classification report.

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F1 Score</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>3090</td>
</tr>
<tr>
<td>True</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>3129</td>
</tr>
<tr>
<td>Accuracy</td>
<td></td>
<td></td>
<td></td>
<td>6219</td>
</tr>
<tr>
<td>Macro avg</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>6219</td>
</tr>
<tr>
<td>Weighted avg</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>6219</td>
</tr>
</tbody>
</table>

Regarding the second confusion matrix with TN = 3085, TP = 3120, FP = 5, and FN = 9, the classifier seems to be performing well, as there are a high number of correctly predicted instances for both positive and negative classes (TP and TN), while the number of misclassifications (FP and FN) is relatively low, as illustrated in Table 2 below.

### Table 2. Confusion matrix.

<table>
<thead>
<tr>
<th></th>
<th>Normal</th>
<th>Attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confusion Matrix</td>
<td>TP = 3085</td>
<td>FP = 5</td>
</tr>
<tr>
<td>Attack</td>
<td>FN = 9</td>
<td>TN = 3120</td>
</tr>
</tbody>
</table>

### 4.4. Synthetic Dataset

In this study, we conducted an extensive evaluation of our proposed model and the GRU method for detecting DDoS attacks in a SDN environment, with a particular focus on low-rate DDoS attacks. In order to collect flow data and simulate various types of DDoS attacks, we used the Ryu controller and the Hping3 tool, respectively.

After applying our proposed model and the GRU method in this environment, we found that our hybrid algorithm outperformed both methods in all evaluation matrices with the same number of epochs (7). This superior performance can be attributed to the excellent loss rate of 0.004 achieved by our hybrid algorithm, which is a measure of how well the deep learning algorithm fits the dataset. In contrast, the GRU method recorded a loss rate of 0.448, indicating a lower fit to the dataset.
The loss rate is a crucial metric to consider when evaluating the performance of a deep learning algorithm, especially in the context of detecting low-rate DDoS attacks. This is because low-rate DDoS attacks can be more difficult to detect due to their subtle nature, and a low loss rate indicates a better fit to the dataset, which ultimately translates to better performance in detecting these types of attacks.

Overall, our results demonstrate the effectiveness of our hybrid algorithm in detecting low-rate DDoS attacks in an SDN environment, and they highlight its potential as a valuable tool in safeguarding against these types of cyber threats. In addition, our findings suggest that our hybrid algorithm may be a promising approach for detecting DDoS attacks in other datasets and environments.

Examination of Figure 13 reveals that our hybrid algorithm surpasses the performance of alternative methods when evaluated on unseen data, exhibiting excellent performance across multiple metrics. Specifically, the algorithm achieves near-optimal values for several key evaluation measures while also minimizing the associated loss, thereby indicating its effectiveness in addressing the problem at hand.

Figure 13. The results of our proposed hybrid method with the GRU method regarding the generated dataset.
To compare the performance of our hybrid approach with the GRU method, we used the receiver operating characteristic (ROC) curve as a measure of effectiveness. The results, shown in Figure 14, indicate that our hybrid algorithm achieved an AUC of 0.9988 while the GRU algorithm had an AUC of 0.7133. This represents a significant advantage for our proposed method, which outperformed the GRU method by more than 40%. Based on the AUC values, it is clear that our approach is superior to the GRU algorithm in terms of accurately distinguishing between normal and attack traffic.

![ROC Curve](image.png)

**Figure 14.** ROC curve related to generated dataset.

### 4.5. Comparison with Other Methods

The model was trained and evaluated on the CICDDoS2019 dataset, which contains various types of DDoS attacks. The results obtained from the proposed hybrid model are shown in Table 3 and Figure 15, which show an accuracy of 0.9981, precision of 0.9996, recall of 0.999, and an F1 score of 0.9993. These results outperformed several existing models such as GRUs [13], CyDDoS [30], DDoSNet [31], Meta-classification [32], Convolutional Neural Networks [33], and Klamn backpropagation NN [34]. The accuracy of the proposed model was higher than all other models, including those that used deep learning techniques. The comparison with the existing models showed that the proposed hybrid model achieved superior performance due to the combination of different deep learning techniques, which enabled the model to capture both spatial and temporal information of the data. The model’s ability to process data in multiple domains and its high sensitivity to anomalies allowed it to outperform the other models. The proposed model’s performance was also enhanced by the use of the CICDDoS2019 dataset, which contains a large number of samples and a wide variety of DDoS attacks.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 Score</th>
<th>ROC</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our Hybrid method</td>
<td>0.9981</td>
<td>0.9996</td>
<td>0.999</td>
<td>0.9993</td>
<td>0.999</td>
<td>0.0081</td>
</tr>
<tr>
<td>GRUs [13]</td>
<td>0.9931</td>
<td>0.9914</td>
<td>0.9856</td>
<td>0.9884</td>
<td>0.997</td>
<td>0.0451</td>
</tr>
<tr>
<td>CyDDoS [30]</td>
<td>0.996</td>
<td>0.997</td>
<td>0.997</td>
<td>0.996</td>
<td>0.998</td>
<td></td>
</tr>
<tr>
<td>DDoSNet [31]</td>
<td>0.99</td>
<td>0.995</td>
<td>0.99</td>
<td>0.99</td>
<td>0.988</td>
<td></td>
</tr>
<tr>
<td>Meta-classification [32]</td>
<td>0.97</td>
<td>0.99</td>
<td>0.97</td>
<td>0.978</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Convolutional Neural Networks [33]</td>
<td>0.954</td>
<td>0.933</td>
<td>0.924</td>
<td>0.928</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Klamn backpropagation NN [34]</td>
<td>0.94</td>
<td>0.9122</td>
<td>0.9749</td>
<td>0.943</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 15. Comparison with other methods.

The results of our evaluation demonstrate that our hybrid algorithm performs exceptionally well and meets our expectations. This is confirmed by the comparison conducted in the same dataset, which shows that the hybrid algorithm consistently outperforms the individual algorithm. Therefore, our hypothesis that the hybrid approach would be superior in most cases has been supported by the data.

5. Conclusions

In summary, our research has demonstrated the effectiveness of employing deep learning methods to detect and protect against DDoS attacks in software-defined networking (SDN) environments. Our proposed hybrid deep learning model, which combines a 1D convolutional neural network (CNN), a gated recurrent unit (GRU), and a dense neural network (DNN), has shown superior performance compared to traditional machine learning algorithms, accurately detecting DDoS attacks and ensuring efficient operation of SDN networks. Notably, our model is particularly adept at identifying low-rate DDoS attacks and can detect both short-term and long-term patterns in the input data.

While our research has produced promising results, it is important to consider the limitations of the study. For example, our proposed model was evaluated on a specific dataset, and further testing on different datasets and network topologies is required to verify its generalizability. Additionally, future research should focus on the implementation of effective mitigation strategies once an attack has been detected.

In conclusion, our research highlights the importance of utilizing deep learning techniques in the detection and defense against DDoS attacks in SDN networks. Furthermore, our findings suggest that our hybrid model can serve as a useful tool in detecting DDoS attacks, ultimately contributing to the overall security and stability of SDN networks. Future research should aim to build on our work by exploring additional strategies for enhancing the detection and response to DDoS attacks in SDN networks.
Author Contributions: Conceptualization, H.E.; Methodology, H.E.; Software, H.E.; Supervision, D.Y.-K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: This work is a part of the Ph.D. thesis titled “Designing a Smart Security Framework for Software Defined Networks” at the Institute of Graduate Studies, Istanbul University-Cerrahpaşa, Istanbul, Turkey.

Conflicts of Interest: The authors declare no conflict of interest.

References


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.