An SDN-based Decision Tree Detection (DTD) Model for Detecting DDoS Attacks in Cloud Environment

Jeba Praba. J[1], R. Sridaran[2]
Department of Computer Applications, Christ College, Rajkot, India[1]
Marwadi University, Rajkot, Gujarat, India[1]
Faculty of Computer Applications, Marwadi University, Rajkot, Gujarat, India[2]

***Abstract*—Detecting Distributed Denial of Service (DDoS) attacks has become a significant security issue for various network technologies. This attack has to be detected to increase the system's reliability. Though various traditional studies are present, they suffer from data shift issues and accuracy. Hence, this study intends to detect DDoS attacks by classifying the normal and malicious traffic. The study aims to solve the data shift issues by using the introduced Decision Tree Detection (DTD) model encompassing of Greedy Feature Selection (GFS) algorithm and Decision Tree Algorithm (DTA). It also attempts to enhance the proposed model's detection rate (accuracy) above 90%. Various processes are involved in DDoS attack detection. Initially, the gureKddcup dataset is loaded to perform pre-processing. This process is essential for removing noisy data. After this, feature selection is performed to select only the relevant features, removing the irrelevant data. This is then fed into the train and test split. Following this, Software Defined Networking (SDN) based DTA is used to classify the normal and malicious traffic, then given to the trained model for predicting this attack. Performance analysis is undertaken by comparing the proposed model with existing systems in terms of accuracy, MCC (Matthew's Correlation Coefficient), sensitivity, specificity, error rate, FAR (False Alarm Rate), and AUC (Area under Curve). This analysis is carried out to evaluate the efficacy of the proposed model, which is verified through the results.**

Keywords—Distributed denial of service attack; greedy feature selection; decision tree algorithm; software defined networking; cloud and decision tree detection

I. Introduction

Cloud computing led to the development of technologies due to its services like resource pooling, measured service, broad network access, rapid elasticity, and on-demand self-service. But, security challenges have a dominant issue in cloud computing development. In cloud computing, security requirements include integrity, availability, privacy preservability, confidentiality, and accountability. Among these, availability has been vital, as the main functionality of cloud computing has been to afford on-demand service at various stages. DDoS (Distributed Denial of Service) and DoS (Denial of Service) have been the fundamental techniques to minimize cloud computing availability. SDN (Software Defined Network) has evolved as a novel network paradigm as the SDN characteristics afford the requirement of flexibility, reliability, and secured future networks that could alter the traditional networks. As SDN possesses abilities like decoupling the control plane from the corresponding DP (Data Plane) and centralized controller, traditional networks could be altered with SDN for easy and early detection of DDoS attacks [1]. Clouds and SDNs explore identical designs with three-layered architecture composing an infrastructure layer and computational resources controlled through a control layer that has been controlled through API (Application Program Interface) by applications in the application layer, as presented in Fig. 1.

SDN has been used to deal with DDoS attacks. However, SDN's issues explore the fact that the overall network gets compromised when a specific controller has been flooded with several attacks. The DDoS attack intends to attack the SDN controller through overflowing FT (Flow Table) in the respective DP, as explored in Fig. 2. Cost and limited memory have made FT in DP to be minimum. Hence, whenever a request with an unknown record in FT arises, DP switches forward all the requests directly to the SDN controller, which checks its FT. If it is a legitimate request, it answers with a legal flow. If the requests received have been more at a specific time, the controller takes more time to look at the FT. In addition, the response also enhances that exhausts the controller's resources and makes it unavailable to deal with legitimate requests. Hence, before forwarding a request to the corresponding DP, it is essential to check if the particular request is normal or an attack. In addition, SDN installation would permit novel security issues [2] in SDN as the centralized SDN controller is the bottleneck and turns out to be an SPF (Single Point Failure) in SDN. Attackers will also concentrate on the DDoS attacks, particularly on the SDN controller for flooding the network traffic in the controller. This controller could no longer reply to the elements of DP, like network routers and switches. This collapses the overall network and makes all the cloud services not accessible to end-users through resource exhaustion, resulting in reputation and economic loss.

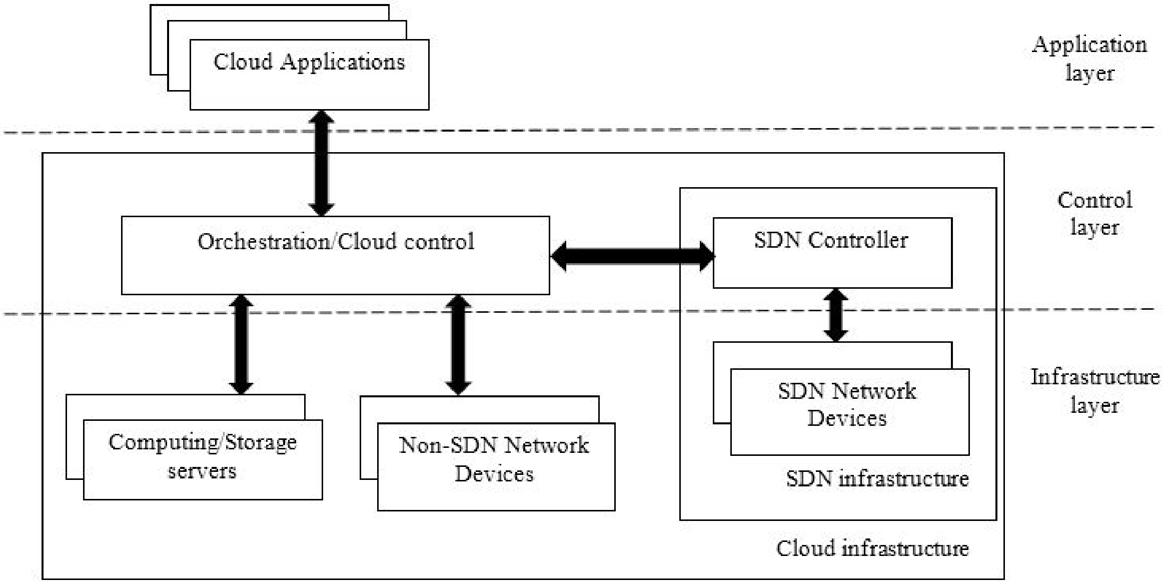


Fig. 1. The Architecture of SDN based Cloud.

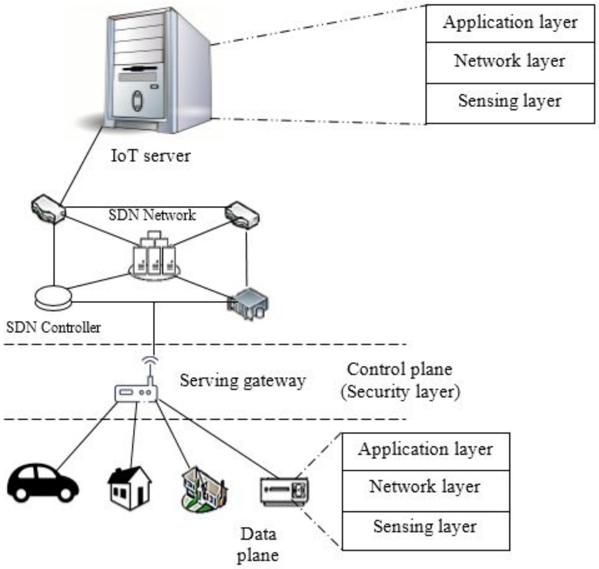


Fig. 2. DDoS Attack Detection in SDN.

Though some efficient studies have been undertaken to detect DDoS attacks in conventional computing environments, these attacks are turning out to be highly occurring in cloud computing environments [3]. The study [4] introduced a design that can detect DDoS attacks in the cloud environment on execution in the real world. Under this design, three ML classification models, RF (Random Forest), LR (Logistic Regression) [5], and SVM (Support Vector Machine), are trained on the CICDDoS2019 dataset. The introduced algorithm chooses the effective classifier for attack prevention, relying on the traffic rate. RF has been found to show better outcomes with 97% accuracy. In addition, a practical and lightweight alleviation method has been introduced for protecting the SDN framework in contradiction to DDoS flooding, confirming an efficient and secure networking environment based on SDN [6]. The proposed system enhanced the DP with a mitigation and classification module to analyze all the new incoming packets and classify all the normal requests from SYN-flood attacks. Subsequently, it performs suitable countermeasures. Simulation outcomes represent that the introduced defending method might effectively deal with DDoS attacks in downstream servers and SDN [7]. The DDoS attack detection in SDN is shown in Fig. 2, which comprises IoT server, SDN controller, serving gateway, data plane, and control plane. Typically, an IoT sensor shall comprise three layers: an application, network, and sensing layer.

The end-user software like email clients and web browsers utilize the application layer. It allows the software to send and receive information and present essential data to the users. The network layer affords the telecommunication resource operations that allow data transfer amongst the systems. PDN (Public Data Network) might provide real-time telecommunication services and have been defined as sub-networks. The sensing layer shows the data type arriving from specific data sources like web services, conventional WSN (Wireless Sensor Networks), and PSNs (Pervasive Social Networks).

Hence, DDoS attack detection is accomplished in SDN, where the SDN controller manages the overall flow control to improve application performance and network management. This platform usually utilizes protocols to inform switches by directing the path for sending the data packets and runs on the server.

Various researches have been carried out to detect DDoS attacks. An approach based on anomaly intrusion has been introduced in the hypervisor layer to minimize the DDoS attacks amongst VMs (Virtual Machines). The evolutionary neural network has been employed to execute the proposed system that integrates PSO (Particle Swarm Optimization) with NN (Neural Network) to detect and classify traffic exchanged amongst VMs. Analysis of the system showed the efficiency in classifying these attacks with high accuracy and a low false alarm rate [8]. However, the dataset used handles only the traffic exchange amongst VMs; hence, traffic arriving from the outside host could be researched in further studies. As this system relied on soft computing methods, a probable future work has to be chosen as the alternative technique to accomplish a high detection rate and minimum computation time. Similarly, ML (Machine Learning) based system has been recommended to detect DDoS attacks. This system makes inferences by the signatures extracted earlier from network traffic samples. Experimentations have been performed through four benchmark datasets, and the outcomes exhibited an accuracy rate of 96.5%. However, the accuracy rate has to be further improved to enhance the detection rate [9], [10]. However, accuracy and dataset shift issues are common in existing methods. Hence, the present study intends to detect DDoS attacks in the SDN-based cloud environment through the proposed Decision Tree Detection (DTD) model.

The objectives of this study are 1) to address the dataset shift issue efficiently through the proposed Decision Tree Detection (DTD) model comprising Greedy Feature Selection (GFS) algorithm and Decision Tree Algorithm (DTA). 2) To enhance the detection accuracy above 90% using the introduced Decision Tree Detection (DTD) model. 3) To evaluate the performance of the proposed system by comparing it with traditional ML methods concerning detection rate, error rate, FAR (False Alarm Rate), specificity, sensitivity, MCC (Matthew’s Correlation Coefficient), and AUC (Area under Curve).

A. *The Significant Contribution of the Proposed Study*

The proposed model contributes to increasing the security in the cloud system with high accuracy. The proposed system employed GFS in the feature selection process, which reduces the overfitting issues, the accuracy range was improvised, and the model training was accomplished faster. The proposed system then used a DTD algorithm for classification, which forms in a tree structure. Further, it breaks down the dataset into smaller subsets, whereas the related decision tree is incrementally developed at the equivalent period. The system also efficiently performs data shifting. Various studies implemented with DTA and GFS in cloud environments were viewed. Therefore, the system can effectively and efficiently detect DDoS attacks in the SDN-based cloud system.

B. *The Motivation of the Proposed Model*

Though various studies have been implemented in the DDoS attack detection over the SDN-based cloud system with the DT approach, they still fail to enhance the accuracy range. The Decision tree provides an efficient approach for decision-making because it results in lay for the problems, thus that all choices might be challenged. Consents to estimate all the conceivable magnitudes of a decision. Affords an architecture for enumerating the standards of results and the possibilities of accomplishing them. Then, the Greedy algorithm results elucidation for small illustration of the complications can be forthright and easy to understand. The greedy feature selection process results in both forward and backward selection; in the proposed model, the backward selection is employed. Thus, these factors motivate implementation of an SDN Based Decision Tree Detection (DTD) Model for Detecting DDoS Attacks in a Cloud Environment.

C. *Paper Organization*

The paper is organized in the following way. Section I explores the SDN-based cloud’s basic ideas in DDoS attack detection. Followed by the existing works related to this context are analyzed in Section II. Then, the proposed DTD model is described in Section III. The results obtained from the proposed system are discussed in Section IV. Finally, the overall study is summarized in Section V.

II. Review of Existing Work

Various existing studies that correspond to DDoS (Distributed Denial of Service) attack detection in SDN-based cloud are analyzed, and the outcomes are presented. The common problems encountered in these existing systems during this review are also explored in this section.

The SDN (Software Defined Networking) encompasses various management, control, and configuration functionalities from the server. This SDN is partitioned into the control and DP method. From the recent data centers that maintain massive data every moment with servers possessing large data volumes, the SDN driver has increased. Components needed in this technique are costly. This process also consumes more time, and configuration turns out to be manual. Through central and management control, this issue has been solved in SDNs. However, SDN has a weak structure in various threats where DDoS attacks dominate. The main recent disruptions in all security systems have been because of DDoS attacks. These attacks mainly intend to collapse the user’s access path to another server or network source. It occurs by integrating the server and host. Because of this, resources like CPU, memory, and traffic disappear from the host. Hence, this issue occurs when data transfer happens between authentic users and servers. Supplementary attacks could be resolved through rebooting. However, this merging model or flooding is complex. The DDoS attack detection has been analyzed based on entropy in SDN for detecting and controlling the impact of the SDN controller. DDoS attack traffic has been incorporated into typical traffic by a setup of twenty-five and fifty percent of traffic intended towards a host in the SDN network. Simulation has been carried out, and the outcomes explore the threshold value selected to find an effective DDoS attack. Future studies include simulating chi-square to find the attack traffic incorporated with the typical traffic [11]. Similarly, a method based on Inf_{dis} (Information distance) has been used to detect this attack in SDN based cloud environment. Subsequently, ABA (Adaptive Boosting Algorithm) framework has been employed with SDN features to detect DDoS attacks. Finally, experimentations revealed the effectiveness of ABA in detecting this attack in SDN based cloud. Despite various merits in utilizing SDN in the cloud, it makes the cloud system susceptible to multiple novel security attacks like FTO (Flow-Table Overloading) DDoS attacks [12]. The FTS (Flow Table Sharing) method has been used to protect SDN-based clouds from FTO DDoS attacks to prevent this attack. This method uses idle FT of supplementary OFS (Open Flow Switches) in-network to protect the FT of switches from overloading. The proposed method enhances the cloud system's resistance against DDoS attacks with minimum engagement of the SDN controller. This leads to minimum communication overhead. The proposed approach has been highly supported by many experiments based on simulation. This shows the efficacy of the proposed system. It is also significant to classify abnormal and normal traffic [13].

Ensemble classification methods comprising SVM (Support Vector Machine), ELM (Extreme Learning Machine), and K-Nearest Neighbour (K-NN) have been suggested for the detection of DDoS attacks through the classification of the traffic as abnormal and normal. The analytical results explored that the proposed K-NN shows an accuracy rate of 76.9%. ELM and SVM classifier performs less or more identical to one another, with an accuracy of 96.4% and 92.7%. The overall decision is undertaken through a max-vote methodology [14], [15]. Various existing methods used different techniques for detecting this attack. The distributed blockchain method has been employed to detect and prevent DDoS attacks on SDN's centralized control plane. The proposed system has been simulated through the use of the AnyLogic simulator. The outcomes revealed the efficiency of the introduced system more than traditional systems, as it adds only minor overhead. Results explored that the controller's overhead was minimized up to thirty-five percent. This also substantially minimized the SDN controller's DDoS attack risk and overhead. A HIDS (Host-based Intrusion Detection System) has been presented to monitor the intrude's activity. The host machine would permit the administrator to monitor the attacker and their activities and alert the data owner in the cloud [16]. The proposed method has enhanced efficacy over the overall system's performance [17], [18].

Similarly, TEHO-DBN (Taylor Elephant Herd Optimization-Deep Belief Network) has been used to detect DDoS attacks in the cloud computing environment. This proposed classifier determines if the particular user is normal or an attacker. Simulation has been undertaken, and it could be summarized that the introduced TEHO relying on DBN has enhanced the performance with an accuracy of 83%. Though the accuracy is better, it has to be further improvised for efficient detection of DDoS attacks [19]. Hence, a Bi-fold SDN-based solution has been recommended using a covariance matrix and genetic algorithm (GA). Traffic data (real-time) has been gathered from an analyzer tool named Tshark network. The Bi-fold method has been employed to distinguish the abnormal traffic. GA takes an initial decision regarding the abnormal and normal attacks. The covariance matrix has been used for refining decisions. Empirical outcomes confirmed the efficiency of the introduced method with better sensitivity, specificity, and accuracy. But, the consumption of time to detect attacks is higher, but it is tolerable simultaneously. In addition, minimizing biased data is also significant in enhancing the attack detection rate [20].

The article [21] examined all the features extracted from SDN traffic, minimizing bias data from the dataset. The traffic features have been assessed through a tenfold-cross validation method. The efficiency of the proposed dataset has been assessed through comparison with the supplementary dataset, for instance-KDDCUP99 (Knowledge Discovery and Data mining tools Competition) dataset. The outcomes revealed that the introduced dataset could be efficiently used for SVM on SDN. A live traffic analysis technique has been provided with the NN (Neural Network) [22]. The proposed TFC-NN (Traffic Flow Classifier-Neural Network) has been trained by a labeled dataset built from under traffic and regular traffic of SDN. A live reduction process has also been integrated with TFC-NN relying on detecting DDoS. The recommended method has been deployed and assessed on SDN architecture relying on various performance metrics under different scenarios of DDoS attacks. Through TFC-NN, Classification has been accomplished with Global accuracy (96.13%). SDN and fog computing has been integrated as a mitigation method to accomplish better outcomes [23]. It also considers the IP spoof an excellent way to detect DDoS attacks. Proposed IP-spoof detection has been undertaken near the attacker source in this study to enhance the attack trace. In addition, a model has been introduced for detecting and mitigating all the DDoS attacks in the cloud environment [24], [25]. The introduced model needs small storage and the ability for fast detection. Empirical outcomes explored the power of the system to ease many attacks. Processing time and detection accuracy were the performance metrics used to assess the proposed model's performance. From the outcomes, it has been clear that the proposed system accomplished high accuracy of 97% with reduced false alarms [26], [27]. Similarly, issues have been solved by introducing an effective system named Prodefense to detect and mitigate DDoS attacks. It also includes criteria that are application-specific for the corresponding threshold of the network traffic. This allows the execution of customizable measures to detect DDoS attacks [28].

Likewise, a modular and flexible architecture has been suggested to alleviate and detect LR-DDoS (Low Rate DDoS) attacks in SDN-based clouds. Notably, the IDS (Intrusion Detection System) has been trained in the suggested architecture through the use of six ML (Machine Learning) models such as RF (Random Forest), SVM, MLP (Multilayer Perceptron), RT (Random Tree), and J48 to assess their performance through the use of CIC (Canadian Institute of Cybersecurity) DoS dataset. Evaluation findings reveal that the introduced method accomplished a 95% detection rate, irrespective of the complexity of detecting LR-DoS attacks. Simulation has been carried out equivalent to real-world production through the usage of the ONOS (Open Network Operating System) controller that has been running on MVM (Mininet Virtual Machine), which showed better outcome. Fast attack detection is also another significant parameter to be considered [29]. For this purpose, a DDoS attack alleviation architecture has been recommended to combine a programmable network observance to permit flexible controlling structure and attack detection for specific and fast attack reactions. To manage the structure, an attack detection system based on a graphic model has been introduced that could handle the dataset shift issue [30]. Simulation outcomes revealed that the suggested architecture could efficiently and effectively solve the security issues caused by the novel network paradigm. It has also been concluded that the proposed attack detection could efficiently state several attacks through real-world cases. Empirical analysis of ML methods has been carried out for detecting Botnet DDoS attacks [31].

Evaluation has been carried out on KDD99 and UNBS-NB 15 datasets for detecting Botnet DDoS. Typically, ML methods such as SVM, NB (Naïve Bayes), USML (Unsupervised Machine Learning), ANN (Artificial Neural Network), and DT (Decision Tree) have been analyzed concerning FPR (False Positive Rate), AUC (Area under Curve), accuracy, FAR (False Alarm Rate) and MCC (Matthews Correlation Coefficient). Analytical results revealed that the KDD99 dataset's performance was better than UNBS-NB 15. This substantiation has been crucial in network security and other relevant areas [32].

Various problems identified through the review of different existing methods for DDoS attack detection in SDN-based cloud environments are discussed below,

- Only a few parameters have been taken into analysis that encompasses the FPR (False Positive Rate) and attack detection rate [12], [16]. In addition, the current work [14], [20] considered only accuracy, specificity, and sensitivity. The present study considers many performance metrics for comparative analysis, such as detection rate, error rate, FAR (False Alarm Rate), specificity, sensitivity, MCC (Matthew's Correlation Coefficient), and AUC (Area under Curve), which explores the effective analysis of the proposed system.
- The traditional research [14] used K-NN, ELM, and SVM for DDoS attack detection, and the accuracy rate of K-NN was found to be 76.9%, ELM-96.4%, and SVM-92.7%. The existing system [19] used TEHO-DBN to detect this attack, and the accuracy was 83%. In addition, the article [22] accomplished an accuracy rate of 96.13% through the use of the proposed TFC-NN. Similarly, the paper introduced methods like HIDS with an accuracy of 97 per cent. Only two parameters are considered [24]. Likewise, the article [29] used ML-based methods, and accuracy was 95 per cent. However, accuracy has to be improved further in all these cases for efficient DDoS attack detection. Hence, this article intends to improve the detection accuracy through the proposed DTD model. Its efficiency is confirmed through the results.
- The existing work [22] has not executed and deployed the NIDS (Network-Based Intrusion Detection System) in SDN. However, the present study intends to detect DDoS attacks in SDN based cloud environment.
- Introduced model of the traditional systems performs fast execution. But, these works hardly suffer from performance loss regarding dataset shift issues [30] and detection accuracy [31]. The proposed system aims to solve these issues through the proposed DTD model.
- The traditional system [32] aimed to compare the introduced methods with other ML methods through many evaluation metrics in the future. But, the present work performs comparative analysis in terms of several metrics by considering various ML methods, such as SVM, NB, DT, USML and deep learning (DL) algorithm - ANN.
- Though various studies [31], [32], [28], [23] have been implemented in the DDoS attack detection, it fails to focus on the data shifting issues.

III. Proposed Methodology

The research introduced a model named DTD (Decision Tree Detection), comprising two algorithms such as Greedy Feature Selection (GFS) and Decision Tree Algorithm (DTA), to detect DDoS attacks in SDN based cloud environment. Various techniques and methods exist to detect this attack. In the proposed model GFS is employed for feature selection, which reduces the complexity and selects features faster. Followed by this, the classification is performed with DTA that enhance the accuracy range in class. However, all these methods possess common drawbacks. The data shift issues are not efficiently handled, and detection accuracy is also minimal [30]. To resolve this drawback, this study proposed a DTD model comprising two algorithms such as GFS and DTA, where GFS is used to perform feature selection. For this purpose, the gureKddcup dataset is used, which includes 48 features. In addition, DTA (Decision Tree Algorithm) is used for classification. The overall process of the proposed system is presented in Fig. 3.

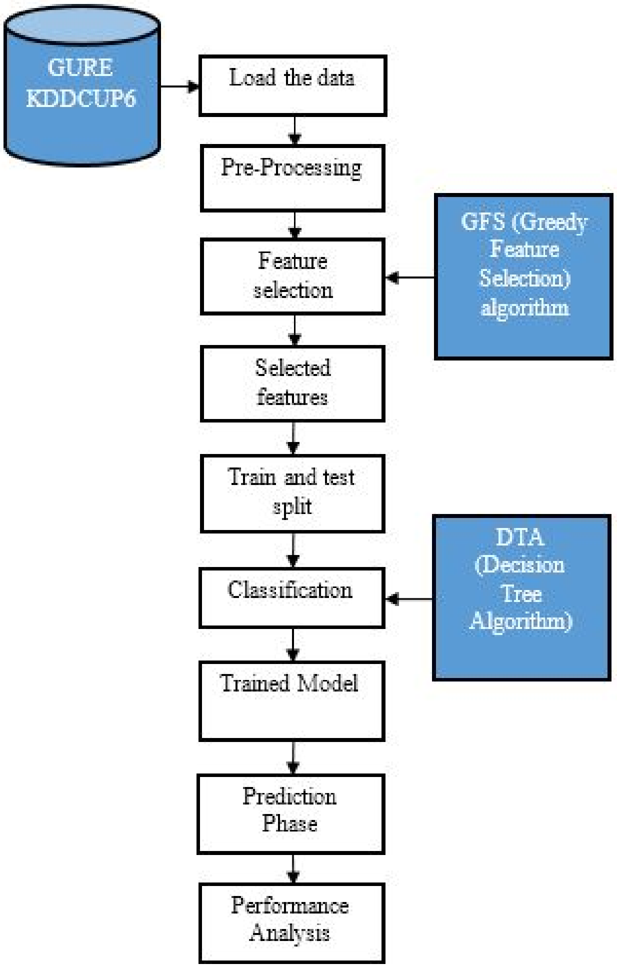


Fig. 3. Dataflow Diagram of the Proposed DTD Model.

Various processes are involved in detecting DDoS attacks. At first, the dataset is loaded, and then pre-processing is performed for noise removal. After this, the feature selection is performed by the GFS algorithm, which filters only five features from all the features available in the dataset; these features are then fed into DTA, one of the efficient classification algorithms that classify non-malicious (normal) and malicious attacks. Here, one portion of the dataset is used for training. The other portion is used for testing. Training data is labeled as local. At the same time, the testing data is labeled as global. After training, the proposed model can detect DDoS attacks effectively, which is proven through results. Finally, performance analysis is undertaken to evaluate the efficiency of the proposed system.

A. GFS (Greedy Feature Selection) Algorithm

GFS is a mathematical method that is simple, easy to implement, and provides solutions to complex issues by making practical decisions [33]. This algorithm operates by recursively building object sets from minimum probable constituent elements. It either selects the best features individually (forward selection) or removes all the worst features individually (backward selection). In the proposed model, the backward selection is employed. The dataset is loaded for pre-processing and feature selection using the GFS algorithm. It performs various steps to accomplish this process, and those steps are presented,

Step 1: Initialize the dataset and its source with the attack features.

Step 2: Generate the objects for the evaluator, search algorithms, and attribute selection.

Step 3: Initiate a Greedy backward search with a filter in accordance with the search algorithms and evaluator over a particular dataset.

Step 4: Measure the error of LOOCV (Leave One Out Cross Validation) of DT classification for the current population set of features of the current search iteration. This is the fitness cost $f(x_{iter})$ for the input set of features of the current iteration x_{iter}.

Step 5: Apply filter to perform greedy operations in a stepwise way. Evaluate it to optimize the search filters.

Step 6: Obtain the count of classes and their attributes. Map the class indexes and update their weight for a pre-defined count of instances.

Step 7: Repeat step 2 to 7 for maximum search iterations.

Step 8: Save and update the minimum fitness cost as $f_{min}(x) = \min f(x)$.

Step 9: Final optimal solution of GFS comprises the significant features from the dataset.

The fitness cost $f(x)$ serves as the parameter to decide on the selection of significant features since it is the evaluation measure of the feature set x. The GFS is executed until the $f_{min}(x)$ is obtained as the optimal solution. Hence, after implementing all these steps through GFS, the dataset comprising 48 columns gets reduced to 5 columns with only relevant and specific features. Feature selection must be made efficiently as it affects the accuracy rate. Implementing GFS for feature selection undergoes the seven steps to select the best features effectively. Finding an issue's solution is typically easier with the GFS algorithm than with other algorithms. Hence, implementing it will enhance the detection accuracy proven through outcomes.

B. DTA (Decision Tree Algorithm)

DTA pertains to the group of SLA (Supervised Learning Algorithms) [34]. It is a tree-based classifier where the internal nodes show the dataset features, individual leaf nodes show the results, and branches show decision rules. Unlike other SLAs, DTA can be used for classification and regression issues. DTs are efficient kinds of algorithms that rely on several learning techniques. It possesses various advantages by boosting the accuracy of prediction models, stability, and straightforward interpretation. DTA is an efficient classification algorithm that exhibits data records into corresponding classes. It utilizes the recursive partition method for data exploration. The DTA components include roots, leaves, and branches. This study pursues a directed tree, which means roots don't have edges. Other components possess a single edge. In addition, the interior nodes show nodes without flow edges. Other nodes are leaves which show the decisional or terminal nodes. The Interior node partitions the space decision into many subspaces based on minimized feature sets. As numeric attributes are considered, attribute spaces are termed conditional ranges. Leaf nodes hold target values to attain their corresponding classes. Under the conditional values, the arrangement of interior nodes occurs from the root node to the leaf nodes. Hence, DTA is used for feature classification based on the below steps.

Step 1: Initialize the tree with a root node (R) that comprises the overall dataset.

Step 2: Determine all the best attributes in the dataset through ASM (Attribute Selection Measure).

Step 3: Determine all the best attributes in the dataset through ASM (Attribute Selection Measure) using information gain I. Information gain is the criterion for estimating the information comprised by each feature attribute. It can be expressed as,

$$I = E_R - (A_R * E_x)$$

Where, E_R is the entropy of the dataset, E_x is the entropy of the feature x, and A_R is the weighted average of the dataset. This measure of entropy helps in the identification of redundant or unnecessary information in an attribute and in the specification of randomness in the data.

Step 4: Generate the node of DT that comprises the best attributes.

Step 5: Recursively make new DTs using dataset subsets developed in step 3. Iterate this process until a particular stage is met, where further classification of nodes cannot be performed. These final nodes are the leaf nodes.

Hence, all the features selected using the GFS algorithm are fed into DTA, which classifies the features based on the above five steps. The attacks are classified as malicious or non-malicious through the overall proposed DTD model. The efficacy of the proposed system is confirmed through the outcomes, which are discussed in the subsequent section.

IV. Results and Discussion

A. Experimental Setup and Dataset Description

The research considered the dataset to evaluate the proposed model for detecting DDoS attacks. It is described in this section. The proposed system also attempts to solve the data shift issue, a common problem in the existing system. A comprehensive analysis of data shift is also explored in this section.

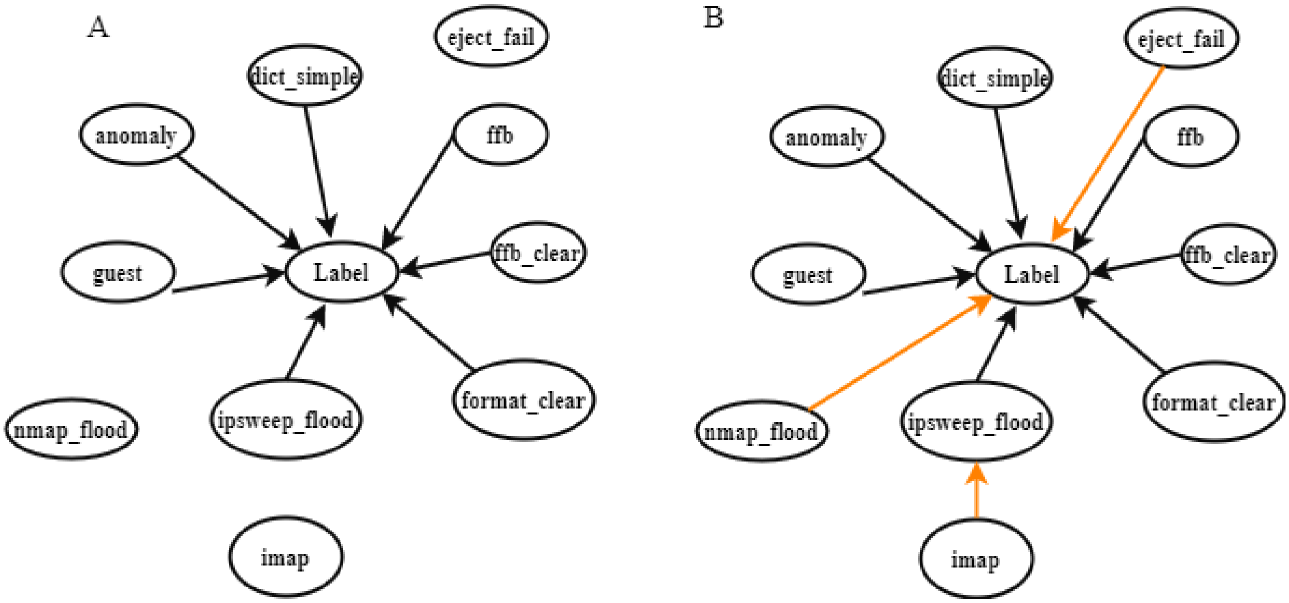
1) Experimental setup: The proposed system is developed and implemented in a system having configurations, as shown below.

Hardware details of the introduced module: Windows 10 pro processor: Intel (R) Core ™ i5-4210U CPU @1.70GHz installed memory, RAM: 4GB, System type: the 64-bit operating system.

2) Dataset description: This study used the gureKddcup dataset [35] to assess the performance of the proposed algorithms, which is one of the widely utilized datasets. It is created regarding the association of Kddcup99 (UCI repository database) and incorporates its payload into individual connections. This dataset helps extract all the information directly from a separate connection's payload to be effectively used in ML processes. In this study, the gureKddcup dataset is used to detect DDoS attacks that consist of 48 attributes which are later reduced to five through the proposed GFS algorithm.

3) Data shift: The data shift issue handles the information association in two subsets of data and assists in predicting a subset, thereby taking into account the data in the supplementary subset. This issue happens when the data generation relies on a model P y1/x1 P(x1), where P(x1) indicates the data distribution or changes amongst the train and test split. It usually occurs when data from a particular class is selected spontaneously compared to a supplementary class. Hence, a large dataset is needed to accomplish high accuracy in this case. During data classification and dataset partitioning in training and testing split, the training dataset is termed Local. The testing dataset is termed Global. It is shown in Fig. 4. As per Fig. 4, it could be seen that a data shift issue occurs in network traffic when a model is constructed using a training dataset. In the proposed method, when new traffic arrives, only some existing data is used as training data (Global). This shows that the proposed detection model keeps updating its training data according to the data received in real-time. Hence, it could always get new observations, afford accurate outcomes, and solve data shift problems. The proposed model nearly functions in real-time and thus solves data shift issues. This infers that the proposed model is not limited or constrained to any particular dataset. To prove the robustness of the model, its performance is compared with the performance of existing datasets.

4) Performance metrics: The performance of the proposed system is analyzed concerning detection rate (accuracy), error rate, specificity, sensitivity, FAR (False Alarm Rate), AUC (Area under Curve), and Matthew's Correlation Coefficient (MCC). Each of the performance metrics is discussed in this section.



(A): Local Update

(B): Global Update.

Fig. 4. Comprising A and B, explores how the Introduced Model Efficiently Handles Data Shift Issues. It is also essential as it affects the Proposed Model's Accuracy. Hence, the Proposed System Efficiently Solves Data Shift Issues, enhancing the Accuracy Rate above 90%.

Accuracy in detecting the attacks can be described as the proportion of detected attacks to the overall attack counts. It is given by the following equation 1.

$$\text{Accuracy} = \frac{\text{Count of detected attacks}}{\text{Overall attack count}} \quad (1)$$

a) Error rate: The error rate is the proportion of attack counts not detected by the overall attack count and is given by equation 2.

$$\text{Error rate} = \frac{\text{Count of attacks not detected}}{\text{Overall attack count}} \quad (2)$$

b) Sensitivity: It is defined as the number of true positives correctly predicted and given by equation 3.

$$\text{Sensitivity} = \frac{\text{True Positive}}{\text{True Positive+False Positive}} \quad (3)$$

c) Specificity: The state/quality of being specific/unique to an individual or a group. It is mathematically represented as per equation 4.

$$\text{Specificity} = \frac{\text{True Negative}}{\text{True Negative+False Negative}} \quad (4)$$

d) AUC (Area Under Curve): It could be stated as the correct curve integral that explores differences in classification and is given by equation 5.

$$\text{AUC} = \frac{1}{2}\left(\left(\frac{\text{True Positive}}{\text{True Positive+False Negative}}\right)+\left(\frac{\text{True Negative}}{\text{True Negative+False Positive}}\right)\right) \quad (5)$$

e) FAR (False Alarm Rate): It is defined as the ratio in which FA occurs in contradiction to true alarms.

f) MCC (Matthew's Correlation Coefficient): A highly static and reliable rate would afford a high score if detection attained good outcomes in all four confusion matrix classes (True Positive, False Positive, True Negative, and False Negative). It is given by equation 6.

$$\text{MCC} = \frac{\text{True Positive}*\text{True Negative}-\text{False Positive}*\text{False Negative}}{\sqrt{\text{(True Positive+False Positive)(True Positive+False Negative)(True Negative+False Positive)(True Negative+False Negative)}}} \quad (6)$$

B. Experimental Results

The proposed DTD model is implemented, and the results are shown in this section. Initially, the dataset is uploaded. The dataset used in the study is gureKddcup. After the dataset is uploaded, the dataset is viewed. The overall count of instances or records is 10000, and attributes (features) are found to be 48. The proposed evolution of the proposed model is stated in Table I with respect to the class and considered performance metrics. Initially, in class normal, the true positive is 0.985, the false positive is 0.043, Precision is 0.999, recall is 0.985, F-measure is 0.992, and the ROC area is 0.972. In the class warezclient, true positive is 0.965; true negative is 0.015, Precision is 0.428, recall is 0.965, F1-measure is 0.593, and ROC area is 0.996. Similarly, in the class dict, the true positive is 0.961, the true negative is 0, Precision is 0.98, recall is 0.961, F1-measure is 0.97, and ROC area is 0.999. Therefore, in class warezmaster, the true positive is 0, the true negative is 0, Precision is 0, recall is 0, F1-measure is 0, and ROC area is 1. For class teardrop, the true positive is 0.982, the true negative is 0, Precision is 1, recall is 0.982, F1-measure is 0.991, and ROC area is 1. In syslog class, the true positive is 0, the true negative is 0, Precision is 0, recall is 0, F1-measure is 0, and ROC area is 0.499. Similarly, in land class, the true positive is 0, the true negative is 0, Precision is 0, recall is 0, F1-measure is 0, and ROC area is 0.499. In the guest class, the true positive is 0, the true negative is 0, Precision is 0, recall is 0, F1-measure is 0, and the ROC area is 0.489. Similarly, in class imap, the true positive is 0, the true negative is 0, Precision is 0, recall is 0, F1-measure is 0, and ROC area is 0.489. In the weighted average class, the true positive is 0.984, the true negative is 0.043, Precision is 0.992, recall is 0.984, F1-measure is 0.987, and ROC area is 0.976.

The confusion matrix is the performance evaluation of ML classification issues. The output could be two or many classes, a table with four varied combinations of actual and predicted values. The confusion matrix for the proposed model is done as shown in Table II. As per Table II, the diagonal values show the correct prediction rate. Finally, the correct and incorrect classified instances are determined as per Table III.

From Table III, the correctly classified instances are 98.42%, and the incorrectly classified instances are 1.58%. In addition, the kappa statistics, MAE (Mean Absolute Error), RMSE (Root Mean Square Error), RAE (Relative Absolute Error), and RRSE (Root Relative Squared Error). The experimental results show that the proposed system shows high accuracy with a low error rate. The proposed system is compared with the existing system to prove the efficacy of the introduced system over other systems, which is explored in the next section.

TABLE I. PERFORMANCE ANALYSIS

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.985	0.043	0.999	0.985	0.992	0.976	normal
0.965	0.015	0.428	0.965	0.593	0.996	warezclient
0.961	0	0.98	0.961	0.97	0.999	dict
0	0	0	0	0	1	warezmaster
0.982	0	1	0.982	0.991	1	teardrop
0	0	0	0	0	0.499	syslog
0	0	0	0	0	0.499	land
0	0	0	0	0	0.489	guest
0	0	0	0	0	0.489	imap
0.984	0.043	0.992	0.984	0.987	0.976	Weighted Average

TABLE II. CONFUSION MATRIX

a	b	c	d	e	f	g	h	i	<--classified as
9627	142	1	0	0	0	0	0	0	a=normal
4	110	0	0	0	0	0	0	0	b=warezclient
2	0	49	0	0	0	0	0	0	c=dict
0	3	0	0	0	0	0	0	0	d=warezmaster
1	0	0	0	56	0	0	0	0	e=teardrop
0	1	0	0	0	0	0	0	0	f=syslog
1	0	0	0	0	0	0	0	0	g=land
2	0	0	0	0	0	0	0	0	h=guest
0	1	0	0	0	0	0	0	0	i=imap

TABLE III. CORRECTLY AND INCORRECTLY CLASSIFIED INSTANCES

Correctly classified instances	98.42
Incorrectly classified instances	1.58
Kappa statistics	0.7281
Mean absolute error	0.0035
Root mean squared error	0.0587
Relative absolute error	34.0313
Root relative squared error	82.7176
Total number of instances	10000

C. Comparative Analysis

The proposed system is analyzed by comparing it with other algorithms concerning seven performance metrics. The existing algorithms considered for analysis include SVM (Support Vector Machine), DT (Decision Tree), NB (Naïve Bayes), ANN (Artificial Neural Network), and USML (Unsupervised Machine Learning). At first, the comparison is made by considering the existing and proposed model's detection rate and error rate over the corresponding input dataset. It is shown in Table IV.

TABLE IV. COMPARATIVE ANALYSIS OF THE PROPOSED AND TRADITIONAL SYSTEMS [30]

Parameter	**Local**		**Global**	
	[30]	**DTD (Proposed work result)**	**[30]**	**DTD (Proposed work result)**
Detection rate	86.56%	88.80%	89.30%	98.42%
Error	13.44%	11.20%	10.70%	1.58%

From the comparative analysis, as shown in Table IV, it is found that the proposed DTD model shows an 88.80% detection rate in local data and an error rate of 11.20%. In comparison, the existing system [30] shows a detection rate of 86.56% and an error rate of 13.44% in local data. The proposed DTD shows a detection rate of 98.42% and an error rate of 1.58% in Global data. In contrast, the traditional system [30] offers an 89.30% detection rate and 10.70% error rate in Global data. It is also graphically presented in Fig. 5. Hence, the introduced model affords more effective accuracy than traditional systems in both local and global data. In real-time, the attackers don't send similar attack patterns each time, which might vary. As the experimental outcomes show the efficiency of the proposed system in testing data with high accuracy, the introduced model can also be utilized in real-time.

In addition, a comparative analysis is undertaken by comparing the proposed DTD model with the existing methods, such as SVM, NB, ANN, DT and USML in terms of accuracy, sensitivity, specificity, FAR, AUC and MCC. The obtained results are shown in Table V and Fig. 6.

The comparative analysis explores that the proposed DTD model shows an accuracy rate of 98.42%, existing SVM shows a 91.55% accuracy rate, DT of 93.30%, NB of 96.74%, ANN of 97.44%, and USML offers 98.08%. From this comparison, the proposed system shows high accuracy rate than the existing systems. The proposed system also shows high AUC, MCC, sensitivity, and specificity than the traditional systems. The FAR of the proposed system is 1.58%, which is minimum than the existing methods, revealing that the introduced system shows only a minimum error rate with high accuracy. Hence, it can be concluded that the presented system is more effective than the traditional systems in terms of all the considered metrics.

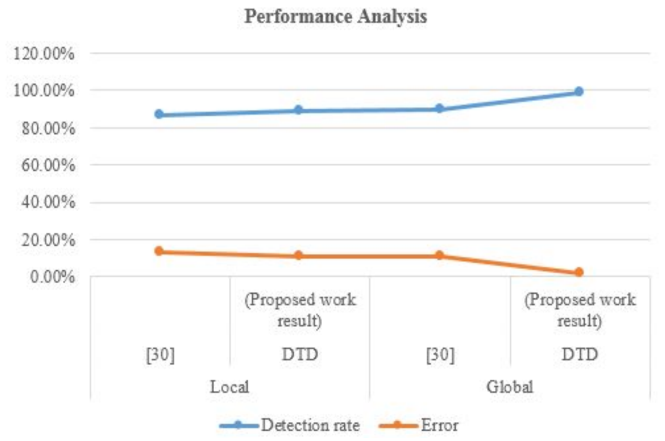


Fig. 5. Comparative Analysis in Terms of Detection Rate and Error Rate [30].

TABLE V. COMPARATIVE ANALYSIS IN TERMS OF VARIOUS METRICS [31]

Performance metrics	**SVM**	**DT**	**NB**	**ANN**	**USML**	**DTD (Proposed work result)**
Accuracy	91.55%	93.30%	96.74%	97.44%	98.08%	98.42%
FAR	8.45%	6.70%	3.26%	2.56%	1.92%	1.58%
Sensitivity	90.13%	93.14%	98.21%	84.89%	91.88%	98.84%
Specificity	9.87%	6.86%	1.71%	15.11%	8.12%	94.30%
MCC	10.46%	5.48%	10.42%	14.46%	1.48%	90.26%
AUC	89.54%	94.52%	89.58%	85.54%	98.52%	98.90%

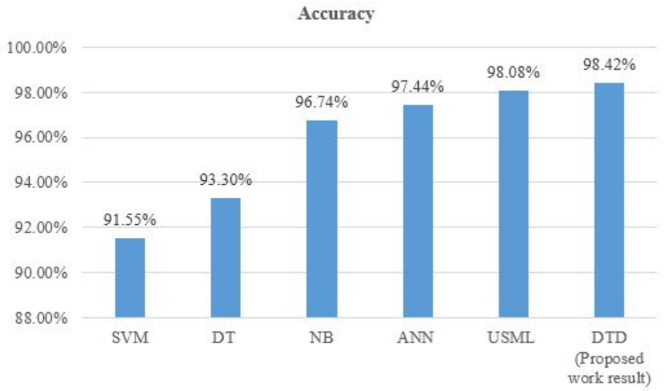


Fig. 6. Comparative Analysis in Terms of Accuracy [31].

V. CONCLUSION

The study detected DDoS (Distributed Denial of Service) attacks through the use of the proposed DTD (Decision Tree Detection) model that is composed of the GFS (Greedy Feature Selection) algorithm for selecting relevant features and DTA (Decision Tree Algorithm) for classifying these features. The proposed model was assessed by comparing it with traditional algorithms such as SVM (Support Vector Machine), DT (Decision Tree), NB (Naïve Bayes), ANN (Artificial Neural Network), and USML (Unsupervised Machine Learning) concerning significant metrics such as accuracy, MCC, sensitivity, specificity, error rate, FAR and AUC. The outcomes explored that the proposed system showed a high accuracy of 98.42% in testing data. As the proposed system showed high accuracy in testing data, it can be employed in real-time and is expected to get efficient results in detecting DDoS attacks. The proposed approach is also more effective than traditional methods, which are confirmed through the outcomes. Hence, these merits show the efficacy of the proposed system in classifying the malicious and normal attacks, thereby efficiently predicting. This results in the proposed model being employed in real-time to enhance the security in the cloud environment. The proposed model is evaluated with only Gurekddcup6 dataset; it can also be evaluated with other datasets. In the future, various other algorithms and hybrid approaches can improve detection accuracy further and perform DDoS attack mitigation.

REFERENCES

[1] F. Shaar and A. Efe, “DDoS attacks and impacts on various cloud computing components,” International Journal of Information Security Science, vol. 7, no. 1, 2018.

[2] K. Bhushan and B. B. Gupta, “Security challenges in cloud computing: state-of-art,” International Journal of Big Data Intelligence, vol. 4, no. 2, pp. 81-107, 2017.

[3] Y. Cui et al., “Towards DDoS detection mechanisms in Software-Defined Networking,” Journal of Network and Computer Applications, p. 103156, 2021.

[4] P. O. Prakash, K. Sasirekha, and D. Vistro, “A DDOS prevention system designed using machine learning for cloud computing environment,” International Journal of Management (IJM), vol. 11, no. 10, 2020.

[5] Y. BN, “Preemptive modelling towards classifying vulnerability of DDoS attack in SDN environment,” International Journal of Electrical & Computer Engineering (2088-8708), vol. 10, no. 2, 2020.

[6] S. Mahrach and A. Haqiq, “DDoS Flooding Attack Mitigation in Software Defined Networks,” International Journal of Advanced Computer Science and Applications, vol. 11, no. 1, pp. 693-700, 2020.

[7] T. Ubale and A. K. Jain, “Survey on DDoS attack techniques and solutions in software-defined network,” in Handbook of computer networks and cyber security: Springer, 2020, pp. 389-419.

[8] A. Rawashdeh, M. Alkasassbeh, and M. Al-Hawawreh, “An anomaly-based approach for DDoS attack detection in cloud environment,” International Journal of Computer Applications in Technology, vol. 57, no. 4, pp. 312-324, 2018.

[9] F. S. d. Lima Filho, F. A. Silveira, A. de Medeiros Brito Junior, G. Vargas-Solar, and L. F. Silveira, “Smart detection: an online approach for DoS/DDoS attack detection using machine learning,” Security and Communication Networks, vol. 2019, 2019.

[10] R. Kesavamoorthy and K. R. Soundar, “Swarm intelligence based autonomous DDoS attack detection and defense using multi agent system,” Cluster Computing, vol. 22, no. 4, pp. 9469-9476, 2019.

[11] R. Chaganti, “Review of Distributed Denial of Service Attack Detection Techniques in Software Defined Networking and Cloud Computing.”

[12] S. W. Tufa, M. Mengstie, H. Gebregziabher, and B. R. Babu, “Detecting Ddos Attack Using Adaptive Boosting with Software Defined Network in Cloud Computing Environment,” REVISTA GEINTEC-GESTAO INOVACAO E TECNOLOGIAS, vol. 11, no. 4, pp. 3485-3494, 2021.

[13] K. Bhushan and B. B. Gupta, “Distributed denial of service (DDoS) attack mitigation in software defined network (SDN)-based cloud computing environment,” Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 5, pp. 1985-1997, 2019.

[14] S. Vimala and J. Dhas, “SDN based DDoS attack detection system by exploiting ensemble classification for cloud computing,” International Journal of Intelligent Engineering and Systems, vol. 11, pp. 282-291, 2018.

[15] A. S. Alzahrani, “An Optimized Approach-Based Machine Learning to Mitigate DDoS Attack in Cloud Computing,” International Journal of Engineering Research and Technology, vol. 13, no. 6, pp. 1441-1447, 2020.

[16] J. Vinnarasi and N. Sudha, “Security Solution for SDN Using Host-Based IDSs Over DDoS Attack,” International Journal of Emerging Technology and Innovative Engineering, vol. 5, no. 9, 2019.

[17] T. Pandikumar and T. Belissa, “Distributed Denial of Service (DDOS) Attack Detection in Software Defined Networking with Cloud Computing,” International Journal of Engineering Science, vol. 12685, 2017.

[18] H. Cheng, J. Liu, T. Xu, B. Ren, J. Mao, and W. Zhang, “Machine learning based low-rate DDoS attack detection for SDN enabled IoT networks,” International Journal of Sensor Networks, vol. 34, no. 1, pp. 56-69, 2020.

[19] S. Velliangiri, P. Karthikeyan, and V. Vinoth Kumar, “Detection of distributed denial of service attack in cloud computing using the optimization-based deep networks,” Journal of Experimental & Theoretical Artificial Intelligence, vol. 33, no. 3, pp. 405-424, 2021.

[20] T. Sindia and J. P. M. Dhas, “A Bifold Software Defined Networking based Defence Mechanism for DDOS Attacks in the Cloud Environment,” International Journal of Applied Engineering Research, vol. 12, no. 20, pp. 9467-9474, 2017.

[21] M. M. Oo, S. Kamolphiwong, T. Kamolphiwong, and S. Vasupongayya, “Analysis of Features Dataset for DDoS Detection by using ASVM Method on Software Defined Networking,” International Journal of Networked and Distributed Computing, vol. 8, no. 2, pp. 86-93, 2020.

[22] O. Hannache and M. C. Batouche, “Neural network-based approach for detection and mitigation of DDoS attacks in SDN environments,” International Journal of Information Security and Privacy (IJISP), vol. 14, no. 3, pp. 50-71, 2020.

[23] K. Sadiq, A. Thompson, and O. Ayeni, “Mitigating DDoS Attacks in Cloud Network using Fog and SDN: A Conceptual Security Framework,” 2020.

[24] M. Zareapoor, P. Shamsolmoali, and M. A. Alam, “Advance DDOS detection and mitigation technique for securing cloud,” International Journal of Computational Science and Engineering, vol. 16, no. 3, pp. 303-310, 2018.

[25] M. A. Aladaileh, M. Anbar, I. H. Hasbullah, Y.-W. Chong, and Y. K. Sanjalawe, “Detection techniques of distributed denial of service attacks on software-defined networking controller–a review,” IEEE Access, vol. 8, pp. 143985-143995, 2020.

[26] N. N. Tuan, P. H. Hung, N. D. Nghia, N. V. Tho, T. V. Phan, and N. H. Thanh, “A DDoS attack mitigation scheme in ISP networks using machine learning based on SDN,” Electronics, vol. 9, no. 3, p. 413, 2020.

[27] Ö. Tonkal, H. Polat, E. Başaran, Z. Cömert, and R. Kocaoğlu, “Machine Learning Approach Equipped with Neighbourhood Component Analysis for DDoS Attack Detection in Software-Defined Networking,” Electronics, vol. 10, no. 11, p. 1227, 2021.

[28] N. Z. Bawany, J. A. Shamsi, and K. Salah, “DDoS attack detection and mitigation using SDN: methods, practices, and solutions,” Arabian Journal for Science and Engineering, vol. 42, no. 2, pp. 425-441, 2017.

[29] J. A. Pérez-Díaz, I. A. Valdovinos, K.-K. R. Choo, and D. Zhu, “A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning,” IEEE Access, vol. 8, pp. 155859-155872, 2020.

[30] B. Wang, Y. Zheng, W. Lou, and Y. T. Hou, “DDoS attack protection in the era of cloud computing and software-defined networking,” Computer Networks, vol. 81, pp. 308-319, 2015.

[31] T. A. Tuan, H. V. Long, L. H. Son, R. Kumar, I. Priyadarshini, and N. T. K. Son, “Performance evaluation of Botnet DDoS attack detection using machine learning,” Evolutionary Intelligence, vol. 13, no. 2, pp. 283-294, 2020.

[32] T. V. Phan and M. Park, “Efficient distributed denial-of-service attack defense in SDN-based cloud,” IEEE Access, vol. 7, pp. 18701-18714, 2019.

[33] I. Tsamardinos, G. Borboudakis, P. Katsogridakis, P. Pratikakis, and V. Christophides, “A greedy feature selection algorithm for Big Data of high dimensionality,” Machine learning, vol. 108, no. 2, pp. 149-202, 2019.

[34] B. Charbuty and A. Abdulazeez, “Classification based on decision tree algorithm for machine learning,” Journal of Applied Science and Technology Trends, vol. 2, no. 01, pp. 20-28, 2021.

[35] Architecture and Technology (KAT/ACT), D.o.U.o.B.C. (EHU/UPV), gureKddcup database, 2019.