I. J. Computer Network and Information Security, 2023, 2, 62-72
Published Online on April 8, 2023 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2023.02.05

LCDT-M: Log-Cluster DDoS Tree Mitigation Framework Using SDN in the Cloud Environment

Jeba Praba. J.*
Department of Computer Applications, Christ College, Rajkot, India
Marwadi University, Rajkot, Gujarat, India
E-mail: prabajjg@gmail.com
ORCID iD: https://orcid.org/0000-0002-3190-8674

R. Sridaran
Faculty of Computer Applications, Marwadi University, Rajkot, Gujarat, India
E-mail: sridaran.rajagopal@gmail.com
ORCID iD: https://orcid.org/0000-0001-7397-7611

Received: 13 October 2022; Revised: 14 December 2022; Accepted: 20 January 2023; Published: 08 April 2023

Abstract: In the cloud computing platform, DDoS (Distributed Denial-of-service) attacks are one of the most commonly occurring attacks. Research studies on DDoS mitigation rarely considered the data shift problem in real-time implementation. Concurrently, existing studies have attempted to perform DDoS attack detection. Nevertheless, they have been deficient regarding the detection rate. Hence, the proposed study proposes a novel DDoS mitigation scheme using LCDT-M (Log-Cluster DDoS Tree Mitigation) framework for the hybrid cloud environment. LCDT-M detects and mitigates DDoS attacks in the Software-Defined Network (SDN) based cloud environment. The LCDT-M comprises three algorithms: GFS (Greedy Feature Selection), TLMC (Two Log Mean Clustering), and DM (Detection-Mitigation) based on DT (Decision Tree) to optimize the detection of DDoS attacks along with mitigation in SDN. The study simulated the defined cloud environment and considered the data shift problem during the real-time implementation. As a result, the proposed architecture achieved an accuracy of about 99.83%, confirming its superior performance.

Index Term: DDoS Attack, Software Defined Networks, Cloud Security, Threat Detection, Log-Cluster DDoS Tree Mitigation.

1. Introduction

Cloud Computing aims to afford hosted services on the internet. Recently, cloud computing has gained extensive attention from numerous markets and user communities. Such services are provided by data centers positioned in varied regions worldwide. There is the extensive usage of cloud computing in IT (Information Technology). Nevertheless, such services possess several security vulnerabilities for their users. Suitable security technologies have not completely evolved. Due to this susceptibility, several service owners are reluctant to adopt cloud computing. Intruders use the flaws of the cloud models and attempt to access the users' confidential data by imposing attacks on the computer's processing ability. Concurrently, SDN (Software Defined Networking) is used with the cloud to explore new ideas for designing a network in a portable and programmable way. SDN is replacing conventional networking technologies due to numerous advantages, including simplified network management, OPI (Open Programmable Interfaces), easy re-configuration, programmability, and centralized control. Despite all such functions and features, SDN security is still a concern due to configuration faults that can cause serious implications. Its programmability aspects might also make it susceptible to attacks. Due to such adverse impacts, securing SDN-based cloud from attack is challenging that has to be solved [1, 2].

Various attacks target SDN-based cloud platforms wherein DDoS (Distributed Denial of Service) is regarded as the usual attack impacting the services of a cloud computing environment. The DDoS attack could be shown as an attack that takes several vulnerable hosts, termed zombies. It undertakes a susceptible attack on devices. The leading cause of huge DDoS attacks in cloud environments is data extortion. This attack results in a significant security hazard to the cloud computing platform by damaging the servers and can hack the information from many users and deter the traffic from attackers. The typical operation of the cloud gets affected by the generation of several attackers called a botnet. The attacker employs a botnet to divert traffic, thereby breaching confidential information. Undesirable traffic situations generated by hackers create pressure on cloud servers to release suitable cloud resources for users [3]. Protecting the

cloud from DDoS attacks has become challenging [4] due to their distributed nature. It has been noticed that the attackers use various DDoS attacks, namely protocol attacks, application layer attacks, and voluminous attacks. It has also become complex to differentiate authentic web traffic from the requests that are a part of DDoS attack and seems obscure in the network. However, conventional research has endeavored to detect DDoS attacks.

Traditional studies have used different methodologies for DDoS detection. The study [5] suggested FT-EHO (Fuzzy and Taylor-Elephant Herd Optimization) for detecting DDoS attacks. FT-EHO has used a fuzzy classifier to learn the rules. Comparison has been undertaken with conventional techniques. Outcomes have explored that, suggested FT-EHO has explored maximum outcomes with a 97.2% detection rate. A comprehensive assessment of current attack detection methodologies has explored that applying ML (Machine Learning) methodologies for detecting DDoS attacks in the cloud is promising work [6]. Besides detection, the research [7] has intended to focus on DDoS attack prevention problems. It has been exposed that there is a significant enhancement in existing methods. However, existing techniques have to be strengthened to reduce this issue. Following this, agglomerative clustering, K-means algorithm, and DBSCAN (Density-Based Spatial Clustering of Applications with Noise) have been used. Experimental outcomes have revealed that clustering with minimum significant features explored better outcomes. Compared with standard PCA, the agglomerative method has exposed a high adjusted rank index rate of 0.9130 [8, 9].

Though existing works have tried to detect DDoS attacks, they have been deficient regarding detection rate and dataset shift issues. Motivated by such drawbacks, the present study attempts to resolve them by proposing a framework to detect and mitigate DDoS attacks in cloud-based SDN environments as per the objectives below. Moreover, in soft-computing attack mitigation methodologies, the classification techniques attain specific focus due to quick response. Moreover, when integrated with the feature selected methodology, the classification approach tends to accomplish the maximum detection rate of DDoS attacks. While utilizing such methods, few parameters are related to classification for managing the learning process. The present study also contributes to assessing the performance of tree-based mitigation of DDoS attacks. This assists in evaluating the impact of learning from the former traffic to sequential traffic. Such DDoS mitigation helps averts malign traffic from reaching its corresponding target by restricting the influence of attack, thereby permitting normal traffic to get through for usual business. With this mitigation, the end users will benefit from consistent service. It will also help business organizations to afford consistent and reliable customer service.

The significant contributions of this study are:

- To perform DDoS attack detection in an SDN-based cloud platform by addressing the dataset shift problem using the proposed LCDT-M (Log-Cluster DDoS Tree Mitigation) framework to mitigate DDoS attacks.
- To detect DDoS attacks in considerable time using the proposed LCDT-M framework.
- To map all the probable correlations within access trace-log patterns and corresponding system activity through the LCDT-M framework.
- To accomplish enhanced accuracy using the LCDT-M framework encompassing GFS (Greedy Feature Selection) to select suitable features, TLMC (Two Log Mean Clustering) for clustering, and DM (Detection and Mitigation) based on DT (Decision Tree) for detecting and mitigating DDoS attack.
- To assess the performance of the proposed system by comparison with conventional ML methods regarding AUC (Area Under Curve), accuracy, FAR (False Alarm Rate), MCC (Matthews Correlation Coefficient), specificity, and sensitivity.

1.1. Paper Organization

The paper is structured as follows: Section 1 explores the basic ideas behind DDoS attack detection on the cloud platform. Section 2 follows with the review of conventional works. Subsequently, the overall proposed work is discussed in section 3. Results attained through the execution of the proposed work are presented in section 4, with the study's overall conclusion in section 5.

2. Review of Existing Work

Existing studies have used different methods to detect and mitigate DDoS attacks on the cloud platform. Those researches are reviewed in this section, along with identifying problems faced by such works in DDoS detection and prevention.

Three sub-systems have been recommended: pre-processing sub-system, AASS (Adaptive Attribute Selection Sub-system), and detection with hindrance system. NSL-KDD dataset has been used that help evaluate the endorsed strategy. It has been found that the integration of MAD-RF (Mean Absolute Deviation-Random Forest) worked better than the traditional method [10]. Hence, MAD-RF has been chosen for evaluation. The suggested method has been handled with UDP, ICMP, and TCP protocols based on DDoS attacks. Outcomes have revealed that MAD-RF has worked better. Early detection of controller attacks has been crucial for magnifying network coverage. Though numerous existing methodologies exist, few studies have focussed on SDN [11]. For detecting, susceptible traffics, supervised ML (Machine Learning) algorithms, namely SVM (Support Vector Machine) and NB (Naïve Bayes), have been used. SVM has explored 82% accuracy rate [12]. Similarly, the article [13] has incorporated programmable network monitoring for permitting attack detection and flexible structure. This allows for a quick and specific reaction to attack. Simulation outcomes have

explored the better performance of the suggested model with an 89.30% as detection rate during testing. However, the detection rate has to be enhanced further.

Further, few studies have evaluated the detection of DDoS attacks by simulation, enabling the mitigation of production networks. A flexible and modular SDN-based framework has been used to detect application and transport layer-DDoS attacks through ML [14] and DL (Deep Learning) models. DL models have been claimed to have explored better detection for all attack kinds [15]. Without disregarding the probable advantages of ML, the article [16] intended to choose the ideal features during the training stage through the ML algorithm. A feedback scheme has been considered for reconstructing the detector when noticing various errors dynamically through the selection of MLP (Multi-Layer Perceptron). Outcomes have revealed that the suggested technique could provide satisfactory detection. Likewise, a feature selection approach has been used to enhance the performance by selecting significant features using IG-RA (Information Gain-Ranker Algorithm) [17]. After choosing suitable features, RF (Random Forest), LMT (Logistic Model Tree), and J48 classifiers have been used to detect DDoS attacks. Empirical outcomes have explored that J48 has afforded enhanced detection compared to LMT and RF.

To enhance the system performance, the study [18] has aimed to minimize the collateral damages instigated at the VM level. The suggested mechanism included a module based on request awareness for reducing collateral damages, attaining request awareness, and CS-IDR (Cuckoo Search-based Identification of Request). Outcomes have explored that the suggested technique has reduced RAM, CPU, overall load, cost, and power consumption. FS-WOA-DNN (Feature Selection-Whale Optimization Algorithm-Deep Neural Network) has been endorsed to effectively mitigate DDoS attacks. Furthermore, homomorphic encryption has secured normal data to improve the suggested model's security. The endorsed algorithm has to be simulated through the MATLAB tool. Experimental tests have revealed 95.35% as a prediction rate [19]. Significant features have been selected for classification through Bhattacharya distance computation to minimize the classifier's training time [3]. TEHO-DBN (Taylor Elephant Herd Optimization-Deep Belief Network) has been considered to modify EHO with the Taylor series. Simulation results have revealed that the suggested DBN and TEHO have enhanced the classification performance with 83% prediction. Further, Bi-LSTM (Bi-directional Long Short Term Memory) and CNN (Convolutional Neural Network) have been used for effective anticipation of DDoS attacks. Findings have revealed that the suggested technique has attained 94.52% accuracy using the CIC-DDoS2019 dataset [20, 21].

A framework has been used to improve the detection rate, including a hybrid ensemble FS (Feature Selection) method encompassing three algorithms: wrapper, embedded, and filter. Selected features have been used for training the ML model to identify and detect attacks. Experimentations have been undertaken with the NSL-KDD dataset. Classification accuracy is 98.92% [22]. In addition, a DLS (Dynamic Learning System) has been suggested for detecting DDoS attacks. CA (Complete Autoencoder) has been used as a base learner for classifying the network traffic. It has been trained with TCP-ICMP attacks. Testing has been undertaken with UDP-TCP-ICMP and UDP-TCP datasets. Varied supervised ML models have been employed on feature-engineered datasets to represent the dataset adaptability for ML under optimal parameter tuning within particular values [23]. Due to such advantages, the study [24] has used SVM and ANN along with USML (Unsupervised Machine Learning), NB (Naïve Bayes), and DT (Decision Tree). Results have revealed that algorithms have different accuracy rates, with SVM showing 91.55%, NB exploring 96.74%, USML showing 98.08%, ANN exposing 97.44%, and DT showing 93.3%. Hence, it has been clear that USML has shown superior performance than other considered algorithms [25].

Packet features that explore DDoS attacks in traffic have been discussed. Initially, the suggested system has extracted header areas of the incoming packet. Then, the system compared the TTL values with the stored IP2HC values. When there seems to be irrelevant matching, the system drops the packet. Finally, the Jensen Shannon divergence concept has been used. The results have shown that the recommended system has achieved 97% accuracy [26]. CRT-RS (Chinese Remainder Theorem-based Reversible Sketch) has been designed to obtain a better detection rate. The traffic records produced by CRT-RS, MM-CUSUM (Modified Multi-Chart Cumulative Sum), which assists protocol independent and self-adaptive detection, have been used that have accomplished 98.57% accuracy.

2.1. Research Gap

Major problems identified through the evaluation of the above conventional works are listed below.

- Though conventional works have used different approaches for DDoS detection, they lacked accuracy. For example, the research [10] has utilized MAD-RF, showing 98.226% accuracy. Similarly, the study [12] recommended SVM, which explored 82% accuracy. Following this, the article [19] employed FS-WOA-DNN showing 95.35% accuracy. Further, FT-EHO used by the study [3] has explored 83% accuracy.
- From the wide literature survey, it has been observed that, the problem in the dynamic environment has rarely been discussed [27]. Also, the data for these dynamic environments cannot be clustered quickly. As a result, the simulated models become slower when clustering the data from the dynamic environments.
- The data shift problem [13] occurs while rebalancing the imbalanced data or due to sampling selection bias issues. It might degrade the performance of ML algorithms while detecting DDoS attacks. Thus, a huge dataset is required to achieve higher accuracy.

A similar Decision Tree Detection (DTD) model for detecting DDoS attacks have been made and tested, resulting in 98.42% accuracy [28]. In the proposed work LCDT-M, a fresh attempt has been made to increase the accuracy further.

The DTD model has a limited scope of only detecting the DDoS attack. As mitigation is significant to enhance network security but cannot mitigate the same. This has become the motivation behind developing the LCDT-M framework, which is explained in the next section.

3. Proposed Methodology

Detecting DDoS attacks are vital for business, as it could assist in securing the network's functioning. Though traditional works have endeavored to detect DDoS attacks, they have been deficient regarding the detection rate. Thus, the present study aims to detect and mitigate DDoS attacks in SDN based cloud environment with enhanced accuracy for which it proposes an LCDT-M (Log-Cluster DDoS Tree Mitigation) framework that comprises constituents that include EF (Evaluation and Filter) and DM (Detection and Mitigation module). The proposed framework, encompasses three major algorithms, namely GFS (Greedy Feature Selection), TLMC (Two Log Mean Clustering), and DM (Detection and Mitigation) based on DT (Decision Tree). In this case, GFS performs feature selection where 48 parameters are reduced to 5. Following this, the study proposes TLMC based on K-means clustering, as it possesses innate advantages like being comparatively effective, easy to understand, robust, and fast. When datasets are discrete, then it affords optimal outcomes. Cluster alters with the re-computation of centroids. Subsequently, DM based on DT is proposed, wherein DT can produce understandable rules. It could also function in high-dimensional data, exploring optimal prediction rates. Due to these advantages, the present research proposes an LCDT-M model encompassing algorithms based on GFS, K-means, and DT to resolve the accuracy and dataset shift issues that conventional works have faced. This proposed model performs the DDoS attack detection and mitigation based on specific processes, as shown in Fig.1.

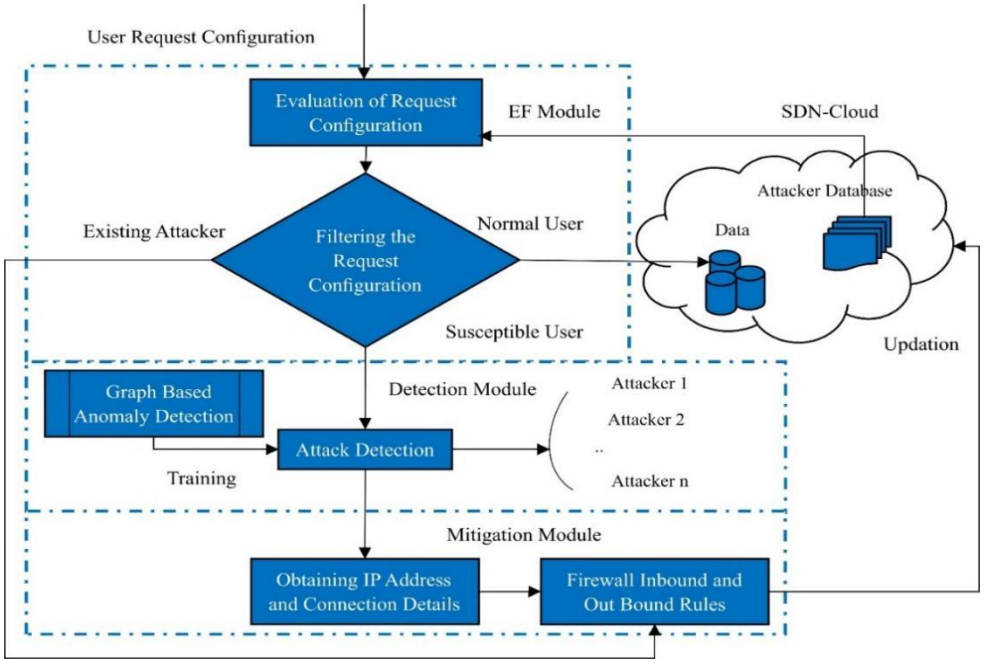

Fig.1. Workflow of the Proposed LCDT-M Framework.

As shown in figure.1, after the user request configuration is received, the module will extract data like IP (Internet Protocol), VP (Virtual Port) addresses, and traffic patterns. The obtained data from the user request configuration is then fed into the EF (Evaluation and filter) module for evaluation. EF component in the framework evaluates the request by comparing it with the attacker database for parameters like IP and VP address. It also monitors the traffic pattern to determine whether the request is from a normal or susceptible user. The request configuration is then moved to filtering based on the type of user. If the user is normal, the user will proceed with data access in the cloud. On the contrary, if the user is susceptible, the request is directed to DM, where the type of DDoS attack is labeled based on anomaly detection using a graph. And if the user is an existing user, then the request is forwarded to the firewall and updated on the attacker's list. To mitigate suspected attacks, it mitigates them by tracing out users' IP (Internet Protocols) addresses, specifically FIOR (Firewall Inbound and Outbound Rules).

3.1. Algorithms for Proposed LCDT-M

The above-mentioned workflow's complex operations involved in the three algorithms are elaborated on in this section.

A. GFS (Greedy Feature Selection Algorithm)

The greedy feature selection (GFS) algorithm [28] selects the features from the gureKddcup dataset [29] by the greedy backward search. First, the dataset is imported into the program and initialized with attack features. Then the generation of objects for attribute selection, evaluator, and search algorithms are done. And it is initiated with the filter based on the evaluator and search algorithm objects over the specified dataset. The filter is applied to perform the step-wise greedy operations and is evaluated for optimization with search filters. The number of attributes and classes in the dataset is obtained from the dataset. These classes are mapped, and their weight sum is updated for a predefined number of instances. The GFS algorithm measures the error of Leave-one-out Cross-validation (LOOCV) within the dataset and identifies the best feature. LOOCV is used here for model validation, and the five features are selected based on the correlation between the data. The GFS ends the loop with the best features selected from the dataset. As a result, the algorithm reduces 48 features into the five best features from the dataset. These five features are based on the grouping made by the GFS as the evaluator and filtration carried out for the input. This is followed by clustering that is undertaken by the proposed TLMC.

B. TLMC (Two Log Mean Clustering Algorithm)

The TLMC used in the study is based on the K-means clustering algorithm and is shown in Algorithm-2. Unlike the conventional K-means clustering, the twice logarithmic mean is calculated here. The selected features from the dataset through the GFS are then fed into the TLMC. The TLMC splits the dataset based on the cluster size. Based on this cluster size, another set of clusters will be formed using the data's logarithmic mean and Euclidean distance. The updated clusters form a perfect cluster as it compares the initial mean value with the updated cluster mean. And the mean value in the updated cluster is estimated to be twice the logarithmic mean value of the split data.

Algorithm-2: Two Log Mean Clustering

```
start
 {
   load_processed_data ()
 {
     data_split(); // data split depends on Cluster Size
    for (i==subject; subject ++;)
    estimate 2*Logarithmic_Mean();
    estimate Euclidean_Distance();
 }
    Cluster_gen (); //new cluster generation
 }
calculate_Mean ();//recalculation of Mean
 end For
 compare_data ()
 {update; //Each Cluster
 }
 generate Perfect_Cluster();
 end
```

C. DM (Detection and Mitigation Algorithm)

The detection and mitigation algorithm are based on the DT (Decision Tree) algorithm, and the overall algorithm is shown in Algorithm-3. DT induction starts with a training process using clustered data and segments at each node, giving the smaller sections. It follows a divide-and-conquer strategy. The dataset contains the attributes of attacks, which are used to identify the type of attack. After segmenting the cluster, the data are grouped into tuples and labeled. The tuple formation is continued until every data is labeled. When any user accesses data from the cloud, the user's external and internal IP addresses can be retrieved from the cloud support of the application interface. Hence, when the attack is identified, external and internal IP addresses get subjected to FIOR, which employs the null-routing method when the attack-site visitors are found to proceed in the way of single or specific destination addresses. Moreover, few routers in the forwarding direction corresponding to attack traffic are configured that abandon the overall traffic towards destinations. Finally, the IP address corresponding to the attack is recorded in the SDN cloud to protect it.

Algorithm-3: Detection and Mitigation

```
Start
ne = Node;
 for (i==ne; ne++;) //Node generation
 {
       All Records (T)
       same attack type;
       return N;
 }
 end For
if (attrbs_available == empty)
{
 get best_attrbs (T, attrbs _available);
 return N;// class with max attack class as leaf node
 attrbs _available = attrbs _available – best_ attrbs;
 split_records (best_ attrbs (best_ attrbs, T))
  }
 for ( Each_split Ti  of T on best_ attrbs)
{
Create a node returned by building LCDT-M;
(Split records Ti, attrbs available)
}
end for
get_IP address (); // get from labelled data
{
Apply firewall_rules (); // firewall rules for outbound and inbound are applied
update SDN_cloud ();
}
end
```

3.2. Mathematical Modelling for LCDT-M

The proposed algorithms, namely GFS, TLMC-based K-means clustering, and DM based on DT, are encompassed within the LCDT-M framework. The mathematical modelling for LCDT-M to detect and mitigate DDoS attacks is discussed below.

Let X be the input dataset containing all the features. This $X$is passed through the greedy backward search filter, which chooses a set of 5 input features heuristically and eliminates the rest for every iteration to obtain the resultant input feature set X_{iter}.

$$X_{iter} = greedybsf(X) \tag{1}$$

The fitness cost $f_{X_{iter}}$of this combination of features X_{iter} is evaluated using the LOOCV (Leave One out Cross Validation) training error.

$$f(_{X_{iter}}) = \frac{1}{n}\sum_{iter=1}^{n}(y_{actual}(x_{iter}) - y_{predicted}(x_{iter}))^2 \tag{2}$$

where n indicates the total number of the data sample, $y_{predicted}(x_{iter})$ is the predicted output and $y_{actual}(x_{iter})$ is the actual output for the input X_{iter}.

As the iteration increases, the fitness cost of the current iteration is compared with that of the previous iteration. The feature combination that gives the maximum error is considered the worst combination, and that combination is eliminated from the search area. The elimination of the feature combination is a way of optimizing the search filter configurations.

$$greedybsfNew = e\lim inate(greedybsf, f_{\max}) \tag{3}$$

where $greedybsfNew$ the new optimized search is filter replacing the old one greedybsf for the next iteration, f_{max} is the feature combination with maximum training error.

Followed by this, the weights for the predefined number of data observations are updated using the equation (4),

$$w^{'}(x_{iter+1}) = w(x_{iter}) - a(\frac{df(x_{iter})}{dw(x_{iter})}) \tag{4}$$

where a is the learning rate, $w^{'}(x_{iter+1})$ is the new weight for the input feature set for the next iteration $iter + 1$ and $w(x_{iter})$ is the current weight of the input feature set at the current iteration iter.

The final minimum fitness cost is updated as the maximum iteration is met.

$$f_{\min}(x) = \arg\min f(x) \tag{5}$$

In equation (5), $f_{min}(x)$ is the final optimal set of features selected by the greedy backward search algorithm of feature selection. This now forms the new dataset X_{new} with which the rest of the processes will be carried on.

The next stage is the TLMC method of clustering the data. The first step is to initialize the number of clusters. In this case, there are two clusters, one representing normal users and another representing attackers. Randomly two centroid points c_1 and c_2 are selected. The Euclidean distance of each data point from the selected centroid is calculated.

$$d_{x,c} = \sqrt{\sum_{i=1}^{5}(x_i - c_i)^2} \tag{6}$$

where x is the data sample, c is the centroid data sample, and i is the feature variable. The data points to their closest centroid are formed as a cluster. The two clusters are a split of the overall dataset.

$$[D_1, D_2] = split(X_{new}) \tag{7}$$

Now the values of the 2-log mean are calculated using the equation (8 and 9) as given below,

$$c_{1,new} = 2 * \log(mean(c_1)) \tag{8}$$

$$c_{2,new} = 2 * \log(mean(c_1)) \tag{9}$$

According to the new centroids, the data points are reassigned to their respective closest centroids. For every iteration, new centroids are created, and reassignments are performed. The terminates when there is no data point left to be reassigned.

The final clusters of data $X_{new,c}$ are then fed into the DM model, where the attack is detected. Initially, a tree with root node R is developed with the $X_{new,c}$. The data $X_{new,c}$ with two clusters of data leave as two branches (subsets) of R. The best attributes are determined through ASM (Attribute Selection Measure) using the information gain index I. This can be evaluated as given in equation (10),

$$I = E_R - (A_R * E_X) \tag{10}$$

where A_R the weighted average of the dataset is, E_X is the entropy of the feature node X, and E_R is the entropy of the dataset. This entropy value helps identify an attribute's redundant or insignificant information. The nodes of DT are then generated with the best attributes.

The error of classification is calculated using equation (11) given below,

$$Error_c = 1 - \max_i[y(i \mid t)] \tag{11}$$

where $y(i|t)$ indicates the data belonging to class i at node t. The new DT nodes are created recursively until the least classification error is obtained. When the network gets fully trained, it is tested with user inputs. When the model identifies that the user is an attacker, the process of mitigating the attack by blocking the corresponding IP address in the firewall is carried out.

4. Results and Discussion

This section discusses the results attained through the proposed system's execution. The experimental setup, dataset

description, and performance metrics are also elaborated in this section with the results of performance and comparative analysis to confirm the effectiveness of the proposed framework in DDoS attack detection and mitigation.

4.1. Experimental-setup and Dataset-description

The dataset used in the study is discussed in this section, along with experimental configurations. As the present research intends to solve the data shift problem, a comprehensive description is given in this section.

A. Experimental-setup

Proposed work is executed through a system with configurations, as shown below:

Hardware details: The program was implemented in a system with Intel (R) Core i5–44422210 CPU @ 3.91 GHz processor, **RAM:** 24GB and a system with 64-bit OS (Operating System).

B. Dataset-description

The research has considered the gureKddcup dataset [29], the latest version of the kddcup99 dataset. Initially, IST (Information System Technology), a Lincoln laboratories group at MIT University, in collaboration with ARFL and DARPA, developed a network. Here, they simulated real traffic with attack and connections, which were sniffed at the Linux command (tcpdump). The experiment was undertaken for seven weeks. After the simulation, connections were extracted from tcpdump files. This was presented in tabular form in the UCI repository. They retrieved 48 attributes for individual connections and class attributes divided into intrinsic, traffic, and content features. This finally resulted in creating kddcup99. As this dataset is older, the newly generated gureKddcup dataset is regarded. This dataset assists in retrieving all the information from distinct connection payloads for effective usage in the ML processes. Thus, the gureKddcup dataset is used for DDoS attack detection, which includes 48 attributes that are then reduced to 5 by the proposed algorithm.

C. Data Shift Issue

The data shift problems could affect the relationship between the output and input data based on the dataset. As discussed earlier, the data shift is considered for the study. While classifying the dataset into training and testing subsets, the training dataset data set is built and referred to as LM local, and the testing set is built and referred to as LM global; here, the LM is referred to as a local machine. While detection is performed on the training dataset, it is referred to as LM local. When any new traffic comes, only a certain portion of prevailing data is utilized as training data and referred to as LM global. This simulation explores that the proposed detection framework should work in a real-time scenario, and it could resolve the data shift issues after the implementation. The detection accuracy rate of the attacks does not affect adversely.

4.2. Performance Metrics

The proposed work has been evaluated regarding metrics, namely AUC (Area under Curve), accuracy, FAR (False Alarm Rate), MCC (Matthews Correlation Coefficient), specificity, and sensitivity. In this study, these metrics are considered as these are standard and suitable metrics typically used to assess the efficacy of the proposed DDoS attack detection and mitigation framework in SDN-Cloud. Due to such significance, various studies [24, 26] have also considered these metrics. Hence, the present research has considered these metrics for analysis.

4.3. Performance and Comparative Analysis

The performance of the proposed system has been evaluated regarding detection rate and error, ROC, knowledge interface, and time complexity. The corresponding outcomes are discussed in this section. The results attained for analysis concerning error and detection rate during the implementation are shown in Table 1.

Table 1. Detection rate and error.

Parameter	Local (Training)			Global (Testing)		
	DaMask [13]	DTD [28]	LCDT-M	DaMask [13]	DTD [28]	LCDT-M
Detection rate	86.56%	88.80%	94.75%	89.30%	98.42%	99.83%
Error	13.44%	11.20%	5.25%	10.70%	1.58%	0.17%

The analysis found that the detection rate of the existing method is 86.56%, DTD has explored 88.80%, while the proposed LCDT-M has shown 94.75% during training. Whereas the existing method has explored 89.30%, DTD has shown a 98.42% detection rate, while the proposed LCDT-M demonstrated 99.83% during testing. In addition, regarding error rate, the existing method has shown 13.44%, while the proposed LCDT-M has shown 5.25% during training. In contrast, the existing method has shown 10.70%, and LCDT-M has exposed a 0.17% error rate during testing. Thus, from the analytical outcomes, it is found that the proposed method's high detection rate and minimum error rate confirmed its efficacy. In addition, ROC analysis has been performed, which is a significant curve reflecting the differences in the proposed model's framework. ROC for the proposed system is shown in Fig.2.

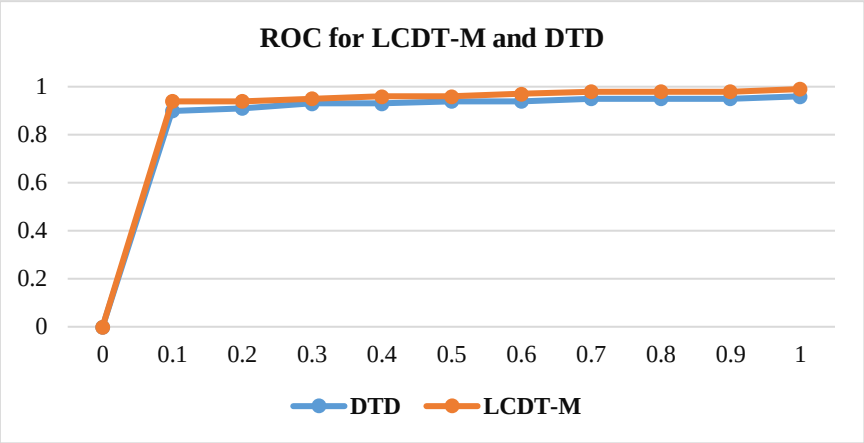

Fig.2. ROC for LCDT-M and DTD.

The ROC value of the proposed LCDT-M is found to be superior to DTD, confirming its efficacy. In addition, analysis has been performed regarding knowledge interference that was recognized over the dataset labels after detection. It was performed over each sample input, and the corresponding values are shown in Table 2.

Table 2. Knowledge interface in LCDT-M.

S.no	Parameters	Values
1	overall performances	400
2	connection	50
3	start time	9
4	orig_port	1
5	resp_port	3
6	orig_ip	5
7	resp_ip	4
8	duration	50

The results show that the displayed attributes are associated with the attack type. Based on the attribute, the type of attack that occurred in the event can be determined. Additionally, the time taken by DTD and LCDT-M for attack detection and mitigation has been compared, and the results are shown in Table 3.

Table 3. Comparison of Time Consumption.

Methods	DTD [28]	Proposed LCDT-M
Time consumption (in ms)	12	8

The obtained outcomes revealed that DTD consumed 12ms, while LCDT-M consumed 8ms. This minimized time consumption is due to the TLMC grounded data clustering process that supports the DT model to recognize the attacker and mitigate the attacks effectively in a shorter duration than the DTD. In addition, the proposed system has been comparatively assessed with conventional works regarding AUC, accuracy, FAR, MCC, specificity, and sensitivity. Different existing methods, namely DT, SVM, USML, ANN, NB, and DTD, have been considered for analysis. Obtained outcomes are shown in Table 4.

Table 4. Comparison of Performance metrics [24].

Performance metrics	DT	SVM	NB	USML	ANN	DTD [28]	Proposed LCDT-M
Accuracy	93.30%	91.55%	96.74%	98.08%	97.44%	98.42%	99.83%
Specificity	6.86%	9.87%	1.71%	8.12%	15.11%	94.30%	98.79%
MCC	5.48%	10.46%	10.42%	1.48%	14.46%	90.26%	98.97%
Sensitivity	93.14%	90.13%	98.21%	91.88%	84.89%	98.84%	99.93%
AUC	94.52%	89.54%	89.58%	98.52%	85.54%	98.90%	99.60%
FAR	6.70%	8.45%	3.26%	1.92%	2.56%	1.58%	0.17%

Analytical results have shown that DTD has a maximum accuracy of 98.42%. However, the proposed method has exposed high accuracy at 99.83%. Likewise, concerning MCC, AUC, FAR, specificity and sensitivity, the proposed system has shown better performance, exploring its effectiveness than conventional methods. Thus, from the performance and comparative analysis, it has been clear that the proposed system has shown more effective performance than conventional methods.

5. Conclusions

The proposed study focused on developing a framework for detecting and mitigating DDoS attacks. It encompassed GFS for feature selection, TLMC for clustering, and DM for detection and mitigation. The results and evaluation of the framework revealed that it could detect and provide mitigation for DDoS attacks in a given environment. Owing to the effective cluster formation and detection of DDoS attacks, the accuracy of the proposed techniques reached about 99.83%, which is better than conventional works. Since the greedy search optimization was used in the feature selection, the time complexity was reduced to 8 ms. Further, the proposed LCDT-M method possesses innate merits, like optimized data filtering and clustering, along with significant feature selection that has made it explore superior outcomes to conventional works. And for the real-time implementation, it was observed that the detection models had a data shift problem, and it was resolved for the proposed technique by the data split as LM local and LM global. The comparative evaluation was also undertaken to confirm the proposed framework's effectiveness, which revealed its superior performance. But the study focused only on DDoS attacks in the networks, which is considered a limitation of the present work. Hence, other types of attacks should also be considered in the future.

6. Declaration

- Conflict of Interest: The Author reports that there is no conflict of Interest.
- Funding: None.
- Acknowledgement: None.

References

[1] B. Alouffi, M. Hasnain, A. Alharbi, W. Alosaimi, H. Alyami, and M. Ayaz, "A systematic literature review on cloud computing security: Threats and mitigation strategies," *IEEE Access*, vol. 9, pp. 57792-57807, 2021.

[2] R. Swami, M. Dave, and V. Ranga, "Software-defined networking-based DDoS defense mechanisms," *ACM Computing Surveys (CSUR)*, vol. 52, pp. 1-36, 2019.

[3] 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, pp. 405-424, 2021.

[4] S. NAIEM, M. I. AMIRA, M. MARIE, E. K. AYMAN, I. GAMAL, H. ABDEL-GALIL, *et al.*, "DDOS ATTACKS DEFENSE APPROACHES AND MECHANISM IN CLOUD ENVIROMENT," *Journal of Theoretical and Applied Information Technology*, vol. 100, 2022.

[5] S. Velliangiri and H. M. Pandey, "Fuzzy-Taylor-elephant herd optimization inspired Deep Belief Network for DDoS attack detection and comparison with state-of-the-arts algorithms," *Future Generation Computer Systems*, vol. 110, pp. 80-90, 2020.

[6] A. Bhardwaj, V. Mangat, R. Vig, S. Halder, and M. Conti, "Distributed denial of service attacks in cloud: State-of-the-art of scientific and commercial solutions," *Computer Science Review*, vol. 39, p. 100332, 2021.

[7] S. Kati, A. Ove, B. Gotipamul, M. Kodche, and S. Jaiswal, "Comprehensive Overview of DDOS Attack in Cloud Computing Environment using different Machine Learning Techniques," *Available at SSRN 4096388*, 2022.

[8] F. J. Abdullayeva, "Distributed denial of service attack detection in E-government cloud via data clustering," *Array*, p. 100229, 2022.

[9] 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, pp. 1985-1997, 2019.

[10] P. Verma, S. Tapaswi, and W. W. Godfrey, "An adaptive threshold-based attribute selection to classify requests under DDoS attack in cloud-based systems," *Arabian Journal for Science and Engineering*, vol. 45, pp. 2813-2834, 2020.

[11] J. Singh and S. Behal, "Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions," *Computer Science Review*, vol. 37, p. 100279, 2020.

[12] A. Mishra and N. Gupta, "Supervised Machine Learning Algorithms Based on Classification for Detection of Distributed Denial of Service Attacks in SDN-Enabled Cloud Computing," in *Cyber Security, Privacy and Networking*, ed: Springer, 2022, pp. 165-174.

[13] 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.

[14] 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.

[15] N. M. Yungaicela-Naula, C. Vargas-Rosales, and J. A. Perez-Diaz, "SDN-based architecture for transport and application layer DDoS attack detection by using machine and deep learning," *IEEE Access*, vol. 9, pp. 108495-108512, 2021.

[16] M. Wang, Y. Lu, and J. Qin, "A dynamic MLP-based DDoS attack detection method using feature selection and feedback," *Computers & Security*, vol. 88, p. 101645, 2020.

[17] A. Patil and D. Kshirsagar, "Towards feature selection for detection of DDoS attack," in *Computing in Engineering and Technology*, ed: Springer, 2020, pp. 215-223.

[18] P. Verma, S. Tapaswi, and W. W. Godfrey, "A request aware module using CS-IDR to reduce VM level collateral damages caused by DDoS attack in cloud environment," *Cluster Computing*, vol. 24, pp. 1917-1933, 2021.
[19] A. Agarwal, M. Khari, and R. Singh, "Detection of DDOS attack using deep learning model in cloud storage application," *Wireless Personal Communications*, pp. 1-21, 2021.
[20] D. Alghazzawi, O. Bamasag, H. Ullah, and M. Z. Asghar, "Efficient detection of DDoS attacks using a hybrid deep learning model with improved feature selection," *Applied Sciences*, vol. 11, p. 11634, 2021.
[21] S. Haider, A. Akhunzada, I. Mustafa, T. B. Patel, A. Fernandez, K.-K. R. Choo, *et al.*, "A deep CNN ensemble framework for efficient DDoS attack detection in software defined networks," *Ieee Access*, vol. 8, pp. 53972-53983, 2020.
[22] N. O. Ogwara, K. Petrova, and M. L. Yang, "Towards the Development of a Cloud Computing Intrusion Detection Framework Using an Ensemble Hybrid Feature Selection Approach," *Journal of Computer Networks and Communications*, vol. 2022, 2022.
[23] M. Aamir and S. M. A. Zaidi, "DDoS attack detection with feature engineering and machine learning: the framework and performance evaluation," *International Journal of Information Security*, vol. 18, pp. 761-785, 2019.
[24] 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, pp. 283-294, 2020.
[25] S. Dong and M. Sarem, "DDoS attack detection method based on improved KNN with the degree of DDoS attack in software-defined networks," *IEEE Access*, vol. 8, pp. 5039-5048, 2019.
[26] 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, pp. 303-310, 2018.
[27] A. Abusitta, M. Bellaiche, and M. Dagenais, "An SVM-based framework for detecting DoS attacks in virtualized clouds under changing environment," *Journal of Cloud Computing*, vol. 7, pp. 1-18, 2018.
[28] Jeba Praba. J, R. Sridaran "An SDN-based Decision Tree Detection (DTD) Model for Detecting DDoS Attacks in Cloud Environment," *International Journal of Advanced Computer Science and Applications(IJACSA)*, vol. 13, pp. 54-64, 2022.
[29] ALDAPA, "gureKddcup and gureKddcup6percent dataset, Computer architecture and technology," University of basque Country 2019.

Authors' Profiles

One of a few researchers in the field of cloud security in the Gujarat province of India, **Mrs. Jeba Praba J.** (Assistant Professor, Dept. of Computer Science & Applications, Christ College, Rajkot, Gujarat, India), is a PhD research scholar of the Marwadi University, Rajkot, Gujarat, India. She has over 17 years of experience in teaching, research, and academic administration. She has been an organizing committee member for various International and National Symposia, Workshops, Seminars, etc. She has published many research papers and book chapters in internationally reputed journals and books. She has also presented her research work at many international and national conferences and has attended several international and national conferences. She is a life member of various International and National Scientific organizations, including the Institute of Electrical and Electronics Engineers (IEEE), the Computer Society of India (CSI) and the International Association of Computer Science and Information Technology (IACSIT), Singapore. She is a highly dynamic teacher, researcher, and academic administrator.

Dr. R. Sridaran is currently working as Dean-Faculty of Computer Applications at Marwadi University, Rajkot, India. He has more than 25 years of experience in the education industry. He is the founder-chairman of the Computer Society of India, Rajkot chapter. His research interests are cloud security, blockchain and networking.

How to cite this paper: Jeba Praba. J., R. Sridaran, "LCDT-M: Log-Cluster DDoS Tree Mitigation Framework Using SDN in the Cloud Environment", International Journal of Computer Network and Information Security(IJCNIS), Vol.15, No.2, pp.62-72, 2023. DOI:10.5815/ijcnis.2023.02.05