International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. #, No.# (Mon-20..)

Security of SDDN based on Adaptive Clustering Using Representatives (CURE) Algorithm

Mohammed Swefee[1], Alharith A. Abdullah [2]

Department of Information Networks, University of Babylon, Hilla, Iraq

E-mail address: mthaaer.net.msc@student.uobabylon.edu.iq, alharith@itnet.uobabylon.edu.iq

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: In the current fact of data center networking, a software-defined data center network (SDDN) has emerged as a transformational solution to address the inherent complexities in network control. Nonetheless, even with so many advantages to look up to, there are critically important issues making its implementation critical, where security, performance, reliability, and fault tolerance are important. For this reason, security becomes a very vital issue, since SDDNs are exposed to many Distributed Denial of Service (DDoS) attacks. In this regard, a new machine-learning-based CURE algorithm framework has been proposed in this paper to outweigh the security challenges. It uses an Adaptive CURE algorithm to minimize the effect of DDoS. The algorithm is designed with adaptive input, depending on the processing resources. The controller captures the suspicious traffic acting as a central coordinator and, if an anomaly in traffic is detected, then the same reforwards a copy of suspicious traffic to the processing and analyzing unit. The adopted approach applies the Adaptive CURE algorithm in processing, through a comprehensive study of the pattern of traffic, the anomalous traffic in the distinguishing of potential DDoS attacks with great accuracy. The algorithm's intelligence facilitates the identification of DDoS attacks. This allows to update switches with suitable flow entries by the controller. Such response mechanisms further improve the security posture of SDDN networks, specifically providing a really strong defense against DDoS attacks. The experiment results show that the proposed framework achieves an accuracy of up to 96.2% with various DDoS attacks.

Keywords: SDDN, DDoS, Datacenter, CURE Algorithm

1. Introduction

The data center in which all infrastructure is virtualized and delivered as a service is known as software-defined data center networking (SDDN) [1]. SDDN allows an inspiring, dynamic, and efficient model for overarching technology innovation in the ruling popular cloud paradigm [2]. One of the common architectures for an SDDN is to use a network virtualization platform that abstracts the underlying physical network infrastructure and allows for the creation of a virtual network stack on top of it, as shown in Fig. 1 [1,3]. Then it could be further configured or controlled by controllers and APIs according to the requirement of this virtual network, allowing more flexibility and chance of automation concerning the data center network. This will separate the control from data planes and allow centralized management and orchestration of the network resources [4,5]. The SDDN is there to provide network simplification administration, including automated provisioning and configuration, so the on-demand ability can be brought up to date in keeping pace with the dynamic changes that always occur in modern data centers. [6,7] The SDDN abstracts network functionality with software, providing programmable infrastructure to enable quick adaptation to changing business requirements and, in so doing, improve operational efficiency and cost reduction [17]. Allowing organizations to benefit from increased network agility, better resource utilization, improved application performance, and, most important, improved security in the data center environment [8].

The rapid evolution of technology has put an ever-increasing reliance on SDDN to delicately handle the advanced and complex challenges presented in modern

E-mail: mthaaer.net.msc@student.uobabylon.edu.iq, alharith@itnet.uobabylon.edu.iq

data center network control. In the course of these new directions, there is a series of pressing questions on particular issues that include security, performance, reliability, fault tolerance [18]. Out of these, security turns out to be a prime focus, whereby DDoS attacks pose great risks to the availability of SDDN infrastructures [5,9].

Generally, the traditional security mechanisms fall short of dynamic and sophisticated DDoS attacks. On the other hand, while inherently centralized, SDDN systems bring many advantages; however, they entail risks by way of posing vulnerabilities that adversaries might exploit [10,11]. Such a single point of failure may be the outcome where central, critical controller management is exposed to some attack. The one single node of a network subject to an attack will cause the failure of the whole network equally suspect to failure [12]. In addition, considering the dynamism and rapidly scalable characteristics of an SDDN environment, the traditional methodologies of mitigation and detection of DDoS may not be very effective in real-time [13-16]. This clearly underlines the need for more advanced and adaptive security solutions, which can effectively help to detect and remediate DDoS attacks in software-defined data center networks.

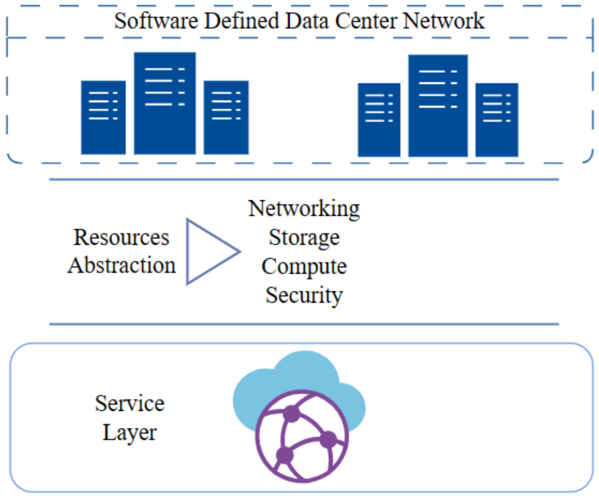

Figure 1. SDDN virtualization

In light of these challenges, this present research tries to discuss and describe the security threats of SDDN associated with DDoS attacks. The proposed framework seeks to bridge the existing gaps in the present defense through a machine learning application, more specifically the CURE algorithm. The CURE algorithm is a bit unlike any other since it includes an innovative way of data point clustering [19]. Cure uses a representative-based strategy, in contrast to the traditional method based on centroids. The selection of a subset of representative points will not allow a strong impact of the outliers and will increase the adaptability of the algorithm under strong diversity and dynamic data patterns. The CURE algorithm is applicable to datasets of high dimensionality, and efficiency comes in through the use of representative points and the random sampling approach [20]. The work principle of the CURE algorithm can thus be summarized in the following section and its outline in Fig. 2 below.

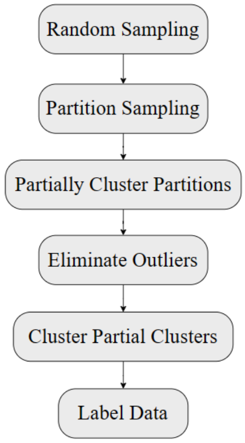

Figure 2. CURE algorithm data clustering steps

This paper comes up with an improved algorithm of CURE, in which the fraction size for random sampling and the representative points are controlled with the processing resources adaptively available. Such an algorithm can efficiently process and analyze vast amounts of data, which would lead to the most precise and on-time detection of DDoS attacks. It fills, therefore, the existing gaps in the traditional defense mechanisms by giving out a solution that is strong and efficient for the ever-growing size and complexity of the modern dataset.

In fact, the existing datasets, such as "CIC-DDoS2019 and UNSW-NB15” [21,22], contribute to an increase in knowledge about security solutions to a great extent. These datasets provide much insight into the nature of DDoS attacks. The above datasets have been analyzed such that security models have been analyzed in such a way that these can be identified and retuned in an effective manner for mitigating the DDoS.

In summary, in this paper:

- We enhanced the CURE Algorithm performance to improve the accuracy of DDoS traffic detection through adaptive resource allocation, this was done by adopting a variable value for the algorithm's inputs depending on the available resources.
- We established a practical testbed simulating the SDDN environment, leveraging the CURE algorithm features using Mininet, and POX controller. This enables the real-time detection of anomalous traffic, particularly indicative of potential DDoS attacks.

The rest of this paper is organized as follows: In the second section we expand on the security part and touch on some researches in this field, then we mention the methods adopted, as well as the problems discussed and solutions. Section 3 reviews present the proposed solution, which is the adaptive CURE-based ML for securing SDDN. Section 4 compares the results with different models of ML algorithms through the related traffic datasets. Finally, we summarized the points as conclusions and included future work on securing the SDDN in section 5.

2. Related Works

Many researchers dealt with Software-Defined Data Center Networks from security aspects [23-25], Kshira Sagar Sahoo et al. addressed the challenge of identifying low-rate DDoS attacks within data center networks based on Software-Defined Networking [26]. Despite the numerous benefits, security concerns within SDN-based data centers, particularly regarding DDoS attacks, remain a focus for research communities. While SDDN is a potent tool for countering attacks, its vulnerability to DDoS attacks is notable, with the logically centralized controller serving as a prime target. Detecting DDoS attacks early is crucial, and this paper introduces a Generalized Entropy (GE) based metric leveraging the flow-based nature of the network to identify low-rate DDoS attacks on the control layer. The proposed detection mechanism, evaluated against Shannon entropy and other statistical information distance metrics, demonstrates improved accuracy. The conclusions focus on the severe effects of low-rate DDoS attacks on the control layer if we assume the detection has not been made in time. An increased rate of packet_in control events would likely cause serious harm, i.e., the depletion of resources in the controller. The research gives a clear indication that GE should be used as an information metric for discrimination among DDoS attacks and normal traffic; hence, it will become possible to use metrics like KLD, Hellinger, and Sibson distance for comparison. This model was used to extract traffic statistics from the flow table through the POX controller in order to give early warnings of potential attacks using the Generalized Entropy metric. On the other hand, the dependency on the traffic patterns, at best circumstances having an 84% accuracy, exerts a major influence on the effectiveness of this method of attack detection and can make it not quite applicable at large to more general and diverse networking environments.

Ammar Moustafa. et al. present a comprehensive approach to addressing security challenges in data center environments using Software-Defined Networking technology [27]. They focus on the need for high throughput, low latency, scalability, and protection from Advanced Persistent Threats (APT) and threats like DDOS attacks. The proposed framework leverages the software-defined network capabilities to bind the network layer to the security middleboxes, such as Intrusion Prevention Systems (IPS) and Firewalls (FW), located at the edge of the network in order to block the attackers. It also hammers on the need for more Out-of-Band security, which helps minimize the latency for services and improves security by blocking detected attackers through Intrusion Detection Systems (IDS). The paper also relates the proof of concept through the implementation using Citrix Xenservers, Mininet Emulation Software, and a real IPS (snort) for simulating a real topology of the data center. Nonetheless, the integration of Security Middleboxes may introduce dependence on definite hardware or software solutions that limit the flexibility of the framework. Performance proves it is able to enhance the security offered by data centers through SDN and reduce the reaction time of attacks. The designed proof of concept to improve security in the SDN-based data centers was completed through the use of Citrix Xenservers, Mininet emulation software, Floodlight controller, and real IPS (snort) to a topology of a real data center. This work proposes an architecture that integrates the network layer with Security Middleboxes, inspecting the security logs and blocking the attacker at the network edge to harden the security of the data center further and hence reduce the response time for the attack.

Raihan UR Rasool. et al. present a novel approach to addressing link flooding attacks in using machine learning techniques [28]. The research focuses on the development of CyberPulse, a system designed to automatically detect and mitigate link flooding attacks. The study highlights the vulnerabilities of software-defined networks to link flooding attacks, particularly on the control channel, and emphasizes the need for effective security measures in this new networking paradigm. By leveraging machine learning algorithms, specifically Multi-Layer Perceptron (MLP), CyberPulse aims to analyze network traffic flow statistics and classify them as either legitimate or flooding traffic. The proposed system consists of modules such as Traffic Flow Statistics, Flood Detection, and Artificial Neural Networks (ANN) Classification, which work together to preprocess traffic data, train the machine learning model, and classify network flows. Through experiments and evaluations, the authors demonstrate the effectiveness of CyberPulse in detecting and mitigating link flooding attacks in real time. The proposed System was implemented by utilizing the Multilayer Perception (MLP) deep learning technique for network traffic classification, forwarding classified results to a Flood Mitigation Module for dropping attack flows using null routing. While the paper contributes to the field of network security by introducing a machine learning-based solution to enhance the security against link flooding attacks with a detection accuracy of up to 80%, The research includes potential scalability challenges and the complexity of integrating and maintaining a machine learning-based system in existing network infrastructures.

Wai-Xi Liu et al. presented a deep learning-based flow classifier for data center networks [29]. Hardly, the classification performances reached by existing classification approaches within the detection latency are proven to be practical and reasonable in real networks: Addressing these limitations, this paper develops a descriptive method by fusing the strength of deep learning for multi-dimensional features with a centralized-controlled, software-defined networking approach. The paper has introduced a fine-grained method of classification of flow using a combination of random-forest-based technology and deep learning models. The model is depending on eight important selected features from the following three dimensions: time distribution features of the flow, real-time features of the flow, and the packet header feature. This paper proposes a flow classification with a two-level architecture and a detection accuracy up to 93.6%. Firstly, pre-classification is built at the level of deep residual learning with AM-Softmax and costs sensitive to detect mice flows. This model is pre-classification applied in deployment on the OpenFlow switch at the network edge, filtering out large amounts of Mice flows. In the second level, deep residual learning with AM-Softmax is deployed on a software-defined networking controller for the extraction of the maximum of information in order to affect an accurate classification of elephant flows. Finally, for the implementation of flow classification over the learning algorithm based on the software-defined data center networks, deep learning is taken using Mininet and RYU controller as the simulation environment on a Linux machine.

Abdulsalam O. Alzahrani et al. study focuses on integrating machine learning algorithms into Software Defined Networking for implementing a Network Intrusion Detection System (NIDS) [30]. The paper highlights the vulnerabilities and advantages of software-defined networks, the impact of network attacks, and the need for real-time systems. It presents the utilization of machine learning algorithms (Decision Tree, Random Forest, XGBoost) for traffic monitoring using the NSL-KDD dataset. Various preprocessing techniques are employed to enhance the effectiveness of the algorithms in attack detection and classification. The study provides a detailed analysis of the dataset, algorithm performance, and comparison with other classical machine learning algorithms. The proposed (eXtreme Gradient Boosting) XGBoost model outperforms other algorithms, with an accuracy of up to 95.95%. l. The paper emphasizes the significance of using limited features for better anomaly detection and highlights the potential for future studies to further enhance anomaly detection using deep neural network algorithms. The implementation tools and environment for the NIDS as application layer based on Machine Learning for Software Defined Networks include the NSL-KDD benchmark dataset for training and testing, feature normalization, feature selection, data preprocessing techniques, and the utilization of XGBoost tree-based machine learning algorithms.

3. Proposed Work

This section explains the structure adopted in building the model and the mechanism for detecting and stopping attacks. the improved version of the proposed CURE algorithm for detecting DDoS attacks in SDDN is also clarified and explains the modifications made to the standard version. We use Mininet with POX controller to simulate the SDDN environment.

A. Proposed Model

In response to the evolving threat landscape of DDoS attacks, the proposed system introduces a robust solution leveraging data mining clustering techniques. Recognizing the limitations of conventional statistical models, our system introduces an innovative approach designed to enhance the detection and prevention of DDoS attacks. The data is processed and apply features selection to eliminate noise and irrelevant information. Then, it was split into the training dataset and the test dataset. However, in the training phase, data training is done using Adaptive CURE as shown in Fig. 3.

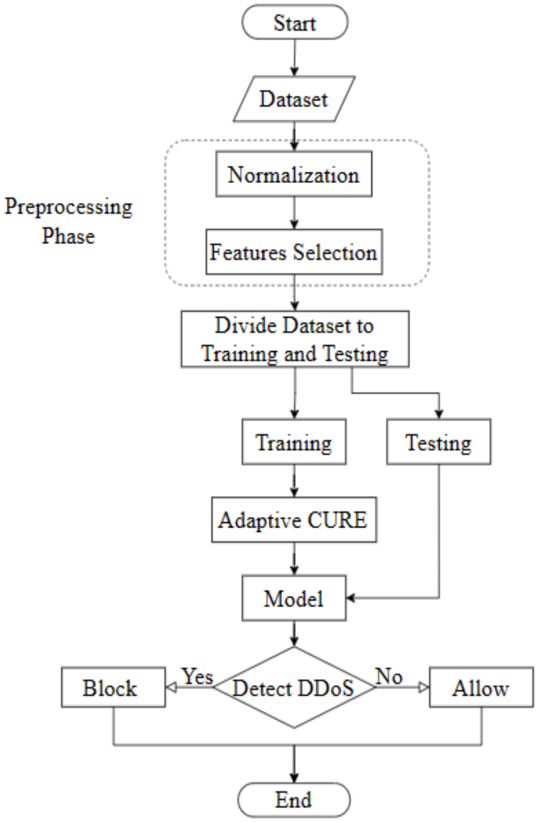

Figure 3. Proposed Model Flowchart

The training phase is centered on the utilization of the Adaptive CURE algorithm through the method of Adaptive CURE Clustering. The used approach is representative-based clustering, suitable in efficiently tackling the occurring problems related to outliers and dynamic data patterns. The algorithm analyzes the training dataset and identifies the clusters. Relevant extracted patterns with representative points were extracted in the field of pattern extraction, seeking to derive useful information on the underlying structure of the network traffic data set and set the clusters boundaries. These patterns encapsulate both normal behavior and DDoS attacks, establishing a foundation for subsequent detection.

B. Network Topology

The proposed framework presents an innovative approach to fortify SDDN networks against DDoS attacks by leveraging the Adaptive CURE algorithm. In building the network, we used the Mininet simulator to prepare the virtual environment and connected the network to a POX controller, which provides software features that allow for packet traffic handling. We used Linux as a server to run the processing operations of the adaptive CURE algorithm.

In this system, upon detecting traffic generated by an attacker aimed at the network, if a flow rule is not found for an incoming packet, the switch forwards a PACKET_IN message containing the header information of the packet to the controller as indicated in Fig. 4. which serving as a central coordinator, it captures the unrecorded traffic and forwards a copy to a processing unit for further analysis. The processing unit then uses the Adaptive CURE algorithm to comprehensively analyze traffic patterns and identify potential DDoS attacks with a high degree of accuracy. Then, the processing unit provides feedback to the controller. Updated with this information, the controller dynamically updates switches with appropriate flow entries, effectively mitigating identified DDoS traffic.

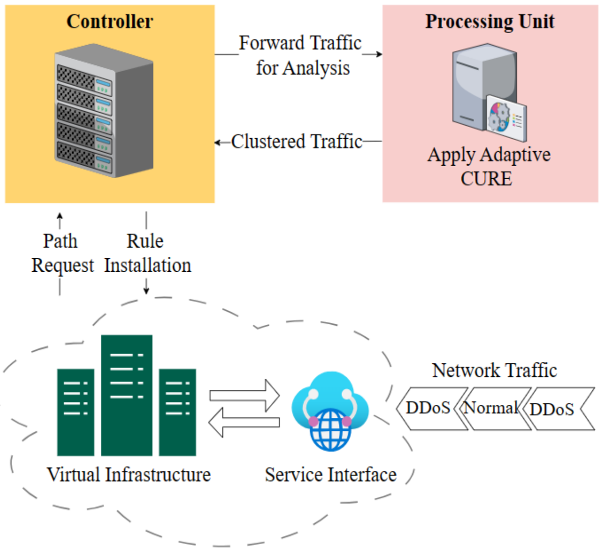

Figure 4. Network Topology

C. Improved CURE Algorithm: Adaptive Resources Allocation

This paper presents an improved version of the CURE algorithm that dynamically adjusts both the fraction size for random sampling and the number of representative points based on available processing resources. The algorithm updates as below:

1) Adaptive Random Sampling: The present and proposed improved CURE method dynamically calculates the fraction size of the adaptive random sampling depending on the available server resources. This kind of step helps improve the efficiency of the algorithm by working with a representative subset of data while keeping in mind the limitations of the server. The algorithm adopt the adaptive fraction size in ensuring that the best use of available resources is made and avoid overload the server. To determine the size of the data fraction with the available memory resources, being careful that it should not fall below 15% or go over 70%, we use the next formula.

$$\text{Fraction}_{\text{Size}} = \max\left(\min\left(\frac{\text{Available}_{\text{Memory}}}{\textit{File size}}, \text{Max}_{\text{Fraction}}\right), \text{Min}_{\text{Fraction}}\right) \quad (1)$$

In this formula: Available Memory represents the available memory resources. Max Fraction is the maximum allowable fraction size (in this case, 70%). Min Fraction is the minimum allowable fraction size (in this case, 15%).

2) Adaptive Number of Representative Points: In addition to adaptive random sampling, the enhanced CURE algorithm leverages available resources to dynamically determine the number of representative points for partially clustered partitions. This adaptive approach ensures that the algorithm strikes a balance between accuracy and computational efficiency. By adjusting the number of representatives based on the process unit resources, the algorithm avoids overloading the process unit. To determine the number of representative points this formula ensures a positive relationship between the fraction size and the number of representative points, and the points increase and decrease proportionally with changes in the fraction size, all while being bounded by the specified minimum and maximum points.

$$\text{No. of Rep. pt} = \max\left(\min\left(\lfloor \text{Fraction}_{\text{Size}} \times \text{Scaling}_{\text{Factor}} \rfloor, \text{Max}_{\text{Points}}\right), \text{Min}_{\text{Points}}\right) \qquad (2)$$

No. of Rep. pt is the number of the representative points. ⌊…⌋ denotes the floor function, rounding down to the nearest integer. Scaling_Factor is the factor that controls the relationship between fraction size and representative points. Max_Points is the maximum allowable number of representative points (e.g., 50). Min_Points is the minimum allowable number of representative points (e.g., 5).

The complete algorithm is represented as Algorithm 1.

Algorithm (1): Adaptive CURE Algorithm

```
Clustering (D, k, f, r)
Input: (D) Dataset, (k) is the number of clusters
Output: (C) Clusters
Begin
    $f \leftarrow \mathrm{Fraction}_{\mathrm{Size}} \leftarrow \max\left(\min\left(\frac{\mathrm{Available}_{\mathrm{Memory}}}{\textit{File size}}, \mathrm{Max}_{\mathrm{Fraction}}\right), \mathrm{Min}_{\mathrm{Fraction}}\right)$ //Calculate fraction size
    $r \leftarrow \max(\min(\lfloor \mathrm{Fraction\_Size} \times \mathrm{Scaling\_Factor} \rfloor, \mathrm{Max\_Points}), \mathrm{Min\_Points})$ // Calculate Number of representative points
    // Initialization: Each Data Point $Pj$ consists of multi-dimension features
    foreach point p in D do
        Set N ← draw a random sample of points (p) based on calculated fraction size (f))
    // end foreach
    $Cluster_{Agglove}C \leftarrow \{Ci \leftarrow \{pi\} | pi \in D\}$ // each point consider a separate cluster with r Number of representative.
    T ← buildkd-tree(D) // T is a data structure to insert the data points
    S← build-heap(D) // S is a data structure to store data points
    While size (S) > k
        u ← extract_min(S)
        v ← u_closest
        delete (S, v)
        m ← merge (u, v)
        delete_rep (T, u); delete_rep(T, v); insert_rep(T, m)
        m.closest ← x // x is an arbitrary cluster in S
        foreach x ∈ S (heap memory)
            if dis (m, x) < dist (m, m.closest)
                m.closest← x
            if x.closest either (u) or (v)
                if dist (x, x.closet) < dist (x, m)
                    x.closest← closest_cluster (T, x, dist(x, m))
                else
                    x.closest← m
                    relocate (S, x)
            else if dist (x, x.closet) > dist (x, m)
                x.closest ← m
                relocate (S, x)
        C ← (S, m)
    Return C // returning (k) closest clusters (C) in S
End
```

4. Results and Discussion

As a well-documented and widely used datasets in the cybersecurity community, the proposed system was tested using CIC-DDoS2019 and UNSW-NB15. The DDoS Evaluation Dataset, known as CIC-DDoS2019, is a comprehensive dataset that resembles real-world data and is designed to evaluate and analyze DDoS attacks; while the raw network packets of the UNSW-NB15 dataset were generated using the IXIA Perfect Storm tool in the Cyber Range Lab of UNSW Canberra [21,22].

In the preprocessing phase of the datasets, a crucial step involves normalization to ensure uniformity in the scale of feature values. This is achieved through the Min-Max normalization technique, expressed mathematically as below:

$$\mathrm{Xnormalized} = \frac{\mathrm{X} - \mathrm{Xmin}}{\mathrm{Xmax} - \mathrm{Xmin}} \tag{3}$$

where X is the original value, Xmin is the minimum value, and X max is the maximum value.

This normalization technique guarantees that each feature lies within the [0, 1] range, mitigating the impact of varying scales on the performance of subsequent machine learning or deep learning models.

Handling empty values is another very crucial part of dataset preprocessing. After detecting missing values in the dataset, in this part, further replacement of values by imputation methods or totally dropping the values takes place. Duplicates, to make sure the data is at least unique, it does so through identification and elimination of duplicate records. Important to mention is that when a duplicate entry is eliminated, it makes the dataset perfect, and therefore, it is considered to be accurate and reliable without introducing any form of bias from the duplication [31]. Normalization can only give a good foundation for clean and standardized data if applied strictly, handling empty values and addressing duplicates, which improves efficiency in the algorithm. We applied feature selection by calculating the mutual information between each feature and the target variable.

$$\mathrm{I}(\mathrm{x}, \mathrm{y}) = \iint \mathrm{p}(\mathrm{x}, \mathrm{y}) \log \frac{\mathrm{p}(\mathrm{x}, \mathrm{y})}{\mathrm{p}(\mathrm{x})\mathrm{p}(\mathrm{y})} \mathrm{dxdy} \tag{4}$$

This process of features selection is crucial for enhancing the efficiency and performance of machine learning models, as it allows for the inclusion of only the most influential features, thereby reducing dimensionality and potential noise in the dataset. The significant features are shown in Table 1 this is after excluding features with similar values.

TABLE I. SELECTED FEATURES FOR ADAPTIVE CURE ALGORITHM

Features	Description
Source port	the transport layer source port
Destination port	the transport layer destination port
Protocol	the protocol types of TCP/IP suite
Flow duration	the duration of the flow
Bwd IAT total	total time between two packets sent in the backward direction
Min packet length	the smallest size observed for captured frames
Down/Up ratio	the ratio between downstream (reception) and upstream (transmission)
Inbound	the direction of data flow

This enhanced the CURE algorithm in terms of accuracy and scalability. It enhances the accuracy of the CURE algorithm by a good margin: adaptive resource allocation is the mechanism that will tune sampling sizes and the number of representative points dynamically. This improvement aims to have an algorithm able to capture fined patterns in the data; thus, it will produce clusters with very high accuracy of representation, as shown in Table 2. In the empirical testing, the improved CURE could give the accuracy of attack detection as high as 96.2%, while the standard algorithm achieved 89.98% on the same dataset. Furthermore, the algorithm is improved to be scalable and suitable for even a wide set of data with different attributes and server configurations. Therefore, its adaptive nature provides ease in scaling toward very effective clustering solutions suitable for large or small-sized applications. For example, as shown in Fig. 5, the results applied to 5000, 15000, and 25000 record-sized data segments show that it was within a ratio of 2.7% of change, this highly outweighs the 8.4% registered in the standard case.

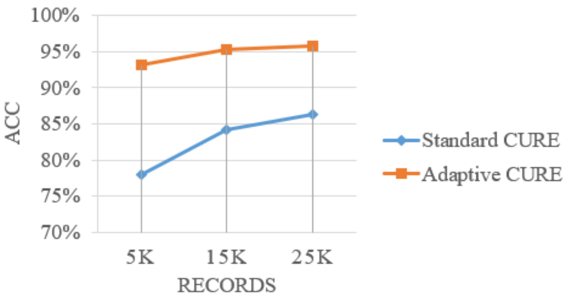

Figure 5. Adaptive CURE Accuracy among Varying Size Dataset

After implementing the practical environment using several algorithms using the same resources and dataset, the results highlighted the effectiveness of Adaptive CURE algorithms in enhancing DDoS attack detection like DNS, NTP, MSSQL, LDAP, and SNMP attacks. Standard CURE demonstrated solid performance across various DDoS attacks, with accuracies ranging from 76.4% to 96.3%. However, Adaptive CURE consistently outperformed the standard version, achieving higher accuracy rates. Adaptive CURE algorithm excelled in detecting DDoS attacks, with accuracies reaching 98.6%. Notably, Adaptive CURE showcased substantial improvement in handling NTP attacks, with accuracies soaring to 90.7%, a marked enhancement compared to Kmeans and the standard version. The results strongly suggest that the Adaptive CURE algorithm significantly enhances DDoS attack detection compared to other algorithms as shown in Fig. 6. The adaptability and improved accuracy of Adaptive CURE make it a promising solution for strengthening cybersecurity measures against evolving cyber threats.

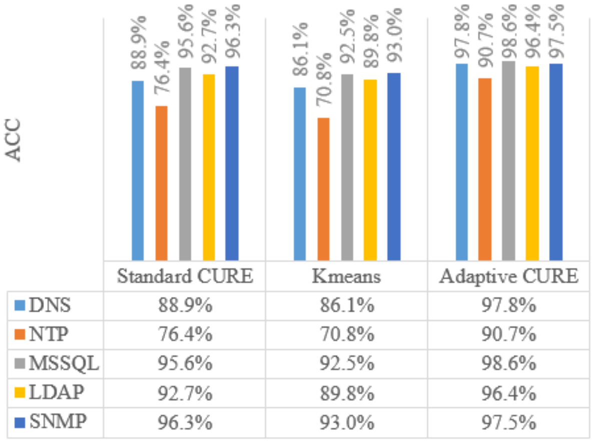

Figure 6. DDoS Detection accuracy using different algorithms

In comparing the results of traffic analysis across various studies using this environment, notable trends emerge as indicated in Fig. 7. Using Generalized Entropy (GE) based metric, Kshira Sagar Sahoo et al. achieved an accuracy rate of 84% [26], while Raihan Ur Rasoo et al. reported a slightly lower rate of 80% using Cyberpulse system with a machine learning-based system utilizing Multi-Layer Perceptron (MLP) algorithms [28]. Wai-Xi Liu et al. findings showed a notable improvement, reaching an accuracy rate of 93.6% using a combination of deep residual learning (DRL) and AM-Softmax [29], followed by Abdulsalam O.

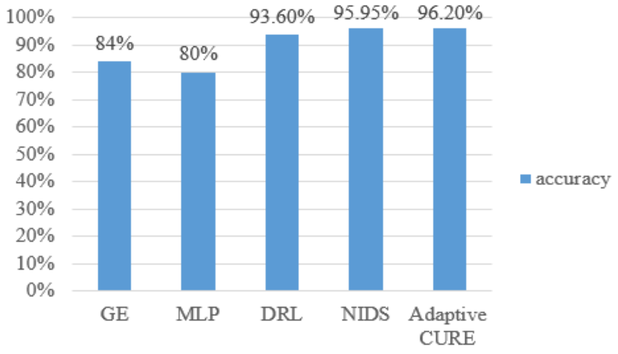

Figure 7. Traffic Analysis Across Various Methods

Alzahrani et al., who achieved an even higher rate of 95.95% by integrating the machine learning algorithms XGBoost to implement a Network Intrusion Detection System (NIDS) [30]. However, the proposed Adaptive CURE system in this study outperformed all mentioned efforts, attaining the highest accuracy rate of 96.2%.

These findings bring out improvements made in DDoS detection where the proposed Adaptive CURE system does bear a lot of significance toward achieving more accuracy and effectiveness in combating DDoS attacks. In these ways, the results of the dataset records processing through Adaptive CURE algorithms presented above underline their remarkable adaptability to evolving DDoS attack patterns, in which they clearly present their effectiveness in dealing with the dynamically changing landscape of cyber threats.

The comparison of performance metrics between Standard CURE and Adaptive CURE clustering algorithms reveals notable differences in their effectiveness for clustering tasks. Table 2 shows that the Adaptive CURE has better performance than the measured parameters in all cases: higher sensitivity, 98.6%, vs. Standard CURE, 95.0%, indicating more capability of giving assurance in recognizing positive cases. Furthermore, Adaptive CURE gives a more precision value of 96.6% compared to Standard CURE with 92.1%, showing really clear capabilities of reducing false positives. The false-negative rate in Adaptive CURE has also achieved a level of 1.3% in comparison to Standard CURE, which is 5.0%, so that contributes to the ability to make the increase in reducing missed detections of positive cases. In general, an accuracy of 96.2% was realized in Adaptive CURE, which was high compared to Standard CURE's 89.8%, thereby proving it to be better in accurate results for clustering. These results point out the effectiveness of an Adaptive CURE for enhancing the performance of the algorithm in clustering. On the other hand, the proposed clustering technique demonstrated noteworthy effectiveness in the recognition of the instances of DDoS attacks.

TABLE II. EVALUATION METRICS RESULTS

Measurement	Standard CURE	Adaptive CURE
Sensitivity	95.0%	98.6%
Precision	92.1%	96.6%
Precision	92.1%	96.6%
FNR	5.0%	1.3%
Accuracy	89.8%	96.2%
F1-Score	93.5%	97.6%
FPR	19.9%	13.1%

5. CONCLUSION AND FUTURE WORK

In the modern scenario of data center networking, the security of Software-Defined Data Center Networks (SDDN) is a very serious issue, specifically the susceptibility of SDDN to Distributed Denial of Service (DDoS) attacks. With the CURE algorithm and bringing its improved version that dynamically adjusts parameters based on available resources, the proposed framework would fill the existing gap in traditional defense mechanisms. The system proposed makes use of a controller that captures suspicious traffic and further forwards a copy of it to a processing unit for analysis using the Adaptive CURE algorithm. It would aid in monitoring all the patterns of traffic thoroughly and thus can very well find and trace the potential attacks of DDoS at a high level of accuracy. From this experiment, the results had shown that it could give up to 96.2% accuracy in detecting different DDoS attacks. Empirical testing proves its efficiency in DDoS attack detection, always outperforming the standard version and other state-of-the-art solutions based on machine learning. It is, therefore, an apparent fact that data set analyses in suchCIC-DDoS2019 and UNSW-NB15 greatly support the proposed framework in shaping the model and optimizing its effectiveness for DDoS detection and mitigation.

Several implications for further research and development can be derived from this paper. Testing the possible integration of the Adaptive CURE algorithm with other machine learning models or clustering algorithms could achieve improved accuracy and reduce the false-positive rate of DDoS attack detection in SDDNs and evaluate the performance using the standalone Adaptive CURE algorithm. Investigate the CURE algorithm for application in alternative network security domains, such as intrusion detection systems and anomaly detection, to investigate how it works and any potential drawback cases. This could be an effort to further seek research and development in this area towards the invention of parallel control planes so that it doesn't encounter single points of failure in its network architecture. This is in line with uninterrupted network operations and, at the same time, increased reliability in the wake of DDoS and other potential threats.

REFERENCES

[1] Darabseh, Ala, Mahmoud Al-Ayyoub, Yaser Jararweh, Elhadj Benkhelifa, Mladen Vouk, and Andy Rindos. "SDDC: A software defined datacenter experimental framework." In 2015 3rd international conference on future internet of things and cloud, pp. 189-194. IEEE, 2015.

[2] Hong, Chi-Yao, Subhasree Mandal, Mohammad Al-Fares, Min Zhu, Richard Alimi, Chandan Bhagat, Sourabh Jain et al. "B4 and after: managing hierarchy, partitioning, and asymmetry for availability and scale in google's software-defined WAN." In

Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, pp. 74-87. 2018.

[3] Raj, Pethuru, and Anupama Raman. Software-defined Cloud Centers. Springer, 2018.

[4] Ammar, Moustafa, Mohamed Rizk, Ayman Abdel-Hamid, and Ahmed K. Aboul-Seoud. "A framework for security enhancement in SDN-based datacenters." In 2016 8th IFIP international conference on new technologies, Mobility and security (NTMS), pp. 1-4. IEEE, 2016.

[5] Sherwin, Jonathan, and Cormac J. Sreenan. "Software-defined networking for data centre network management: A survey." arXiv preprint arXiv:2106.10014 (2021).

[6] Dinh, Phuc Trinh, and Minho Park. "BDF-SDN: A big data framework for DDoS attack detection in large-scale SDN-based cloud." In 2021 IEEE Conference on Dependable and Secure Computing (DSC), pp. 1-8. IEEE, 2021.

[7] Bates, Adam, Kevin Butler, Andreas Haeberlen, Micah Sherr, and Wenchao Zhou. "Let SDN be your eyes: Secure forensics in data center networks." In Proceedings of the NDSS workshop on security of emerging network technologies (SENT'14), pp. 1-7. 2014.

[8] Yang, Zhenjie, Yong Cui, Baochun Li, Yadong Liu, and Yi Xu. "Software-defined wide area network (SD-WAN): Architecture, advances and opportunities." In 2019 28th International Conference on Computer Communication and Networks (ICCCN), pp. 1-9. IEEE, 2019.

[9] Abdelrahman, Abdallah Mustafa, Joel JPC Rodrigues, Mukhtar ME Mahmoud, Kashif Saleem, Ashok Kumar Das, Valery Korotaev, and Sergei A. Kozlov. "Software - defined networking security for private data center networks and clouds: Vulnerabilities, attacks, countermeasures, and solutions." International Journal of Communication Systems 34, no. 4 (2021): e4706.

[10] Meti, Nisharani, D. G. Narayan, and V. P. Baligar. "Detection of distributed denial of service attacks using machine learning algorithms in software defined networks." In 2017 international conference on advances in computing, communications and informatics (ICACCI), pp. 1366-1371. IEEE, 2017.

[11] Prakash, Aditya, and Rojalina Priyadarshini. "An intelligent software defined network controller for preventing distributed denial of service attack." In 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), pp. 585-589. IEEE, 2018.

[12] Badotra, Sumit, Sarvesh Tanwar, Salil Bharany, Ateeq Ur Rehman, Elsayed Tag Eldin, Nivin A. Ghamry, and Muhammad Shafiq. "A DDoS vulnerability analysis system against distributed SDN controllers in a cloud computing environment." Electronics 11, no. 19 (2022): 3120.

[13] Varga, Pál, Georgios Kathareios, Ákos Máté, Rolf Clauberg, Andreea Anghel, Péter Orosz, Balázs Nagy, Tamás Tóthfalusi, László Kovács, and Mitch Gusat. "Real-time security services for SDN-based datacenters." In 2017 13th International Conference on Network and Service Management (CNSM), pp. 1-9. IEEE, 2017.

[14] Mahdi, Suadad S., and Alharith A. Abdullah. "Enhanced security of software-defined network and network slice through hybrid quantum key distribution protocol." Infocommunications Journal 14, no. 3 (2022): 9-15.

[15] Jose, Ancy Sherin, Latha R. Nair, and Varghese Paul. "Towards detecting flooding DDOS attacks over software defined networks using machine learning techniques." Revista Geintec-Gestao Inovacao E Tecnologias 11, no. 4 (2021): 3837-3865.

[16] Sadkhan, Sattar B., Mustafa S. Abbas, Suadad S. Mahdi, and Shahad A. Hussein. "Software-defined network security-status, challenges, and future trends." In 2022 Muthanna International Conference on Engineering Science and Technology (MICEST), pp. 10-15. IEEE, 2022.

[17] Mahdi, Suadad S., and Alharith A. Abdullah. "Improved security of SDN based on hybrid quantum key distribution protocol." In 2022 International Conference on Computer Science and Software Engineering (CSASE), pp. 36-40. IEEE, 2022.

[18] Sabur, Abdulhakim, Ankur Chowdhary, Dijiang Huang, Myong Kang, Anya Kim, and Alexander Velazquez. "S3: a {DFW-based} scalable security state analysis framework for {large-scale} data center networks." In 22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2019), pp. 473-485. 2019.

[19] Kumble, Nikita, and Vandan Tewari. "Improved CURE Clustering Algorithm using Shared Nearest Neighbour Technique." International Journal of Emerging Trends in Engineering Research 9 (2021).

[20] Guha, Sudipto, Rajeev Rastogi, and Kyuseok Shim. "CURE: An efficient clustering algorithm for large databases." ACM Sigmod record 27, no. 2 (1998): 73-84.

[21] Sharafaldin, Iman, Arash Habibi Lashkari, Saqib Hakak, and Ali A. Ghorbani. "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy." In 2019 international carnahan conference on security technology (ICCST), pp. 1-8. IEEE, 2019.

[22] Moustafa, Nour, and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)." In 2015 military communications and information systems conference (MilCIS), pp. 1-6. IEEE, 2015.

[23] Liu, Wai-xi, Jun Cai, Qing Chun Chen, and Yu Wang. "DRL-R: Deep reinforcement learning approach for intelligent routing in software-defined data-center networks." Journal of Network and Computer Applications 177 (2021): 102865.

[24] Li, Jishuai, Tengfei Tu, Yongsheng Li, Sujuan Qin, Yijie Shi, and Qiaoyan Wen. "DoSGuard: Mitigating denial-of-service attacks in software-defined networks." Sensors 22, no. 3 (2022): 1061.

[25] Chung, Chun-Jen, Tianyi Xing, Dijiang Huang, Deep Medhi, and Kishor Trivedi. "SeReNe: on establishing secure and resilient networking services for an SDN-based multi-tenant datacenter environment." In 2015 IEEE International Conference on Dependable Systems and Networks Workshops, pp. 4-11. IEEE, 2015.

[26] Chung, Chun-Jen, Tianyi Xing, Dijiang Huang, Deep Medhi, and Kishor Trivedi. "SeReNe: on establishing secure and resilient networking services for an SDN-based multi-tenant datacenter environment." In 2015 IEEE International Conference on Dependable Systems and Networks Workshops, pp. 4-11. IEEE, 2015.

[27] Ammar, Moustafa, Mohamed Rizk, Ayman Abdel-Hamid, and Ahmed K. Aboul-Seoud. "A framework for security enhancement in SDN-based datacenters." In 2016 8th IFIP international conference on new technologies, Mobility and security (NTMS), pp. 1-4. IEEE, 2016.

[28] Rasool, Raihan Ur, Usman Ashraf, Khandakar Ahmed, Hua Wang, Wajid Rafique, and Zahid Anwar. "Cyberpulse: A machine learning based link flooding attack mitigation system for software defined networks." IEEE Access 7 (2019): 34885-34899.

[29] Liu, Wai-Xi, Jun Cai, Yu Wang, Qing Chun Chen, and Jia-Qi Zeng. "Fine-grained flow classification using deep learning for software defined data center networks." Journal of Network and Computer Applications 168 (2020): 102766.

[30] Alzahrani, Abdulsalam O., and Mohammed JF Alenazi. "Designing a network intrusion detection system based on machine learning for software defined networks." Future Internet 13, no. 5 (2021): 111.

[31] Polat, Huseyin, Onur Polat, and Aydin Cetin. "Detecting DDoS attacks in software-defined networks through feature selection

methods and machine learning models." Sustainability 12, no. 3 (2020): 1035.

Mohammed Swefee received the bachelor's degree in Information Technology from Babylon University, in 2018. He is currently pursuing the master's degree in information Networks with Babylon University. His research interests include software-defined networks, network security, and machine learning.

Alharith A. Abdullah received his B.S. degree in Electrical Engineering from Military Engineering College, Iraq, in 2000. MSc. degree in Computer Engineering from University of Technology, Iraq, in 2005, and his PhD. in Computer Engineering from Eastern Mediterranean University, Turkey, in 2015. His research interests include Security, Network Security, Cryptography, Quantum Computation and Quantum Cryptography