Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

An integrated SDN framework for early detection of DDoS attacks in cloud computing

Asha Varma Songa[1] and Ganesh Reddy Karri[1*]

Abstract

Cloud computing is a rapidly advancing technology with numerous benefits, such as increased availability, scalability, and flexibility. Relocating computing infrastructure to a network simplifies hardware and software resource monitoring in the cloud. Software-Defined Networking (SDN)-based cloud networking improves cloud infrastructure efficiency by dynamically allocating and utilizing network resources. While SDN cloud networks offer numerous advantages, they are vulnerable to Distributed Denial-of-Service (DDoS) attacks. DDoS attacks try to stop genuine users from using services and drain network resources to reduce performance or shut down services. However, early-stage detection of DDoS attack patterns in cloud environments remains challenging. Current methods detect DDoS at the SDN controller level, which is often time-consuming. We recommend focusing on SDN switches for early detection. Due to the large volume of data from diverse sources, we recommend traffic clustering and traffic anomalies prediction which is of DDoS attacks at each switch. Furthermore, to consolidate the data from multiple clusters, event correlation is performed to understand network behavior and detect coordinated attack activities. Many existing techniques stay behind for early detection and integration of multiple techniques to detect DDoS attack patterns. In this paper, we introduce a more efficient and effectively integrated SDN framework that addresses a gap in previous DDoS solutions. Our framework enables early and accurate detection of DDoS traffic patterns within SDN-based cloud environments. In this framework, we use Recursive Feature Elimination (RFE), Density Based Spatial Clustering (DBSCAN), time series techniques like Auto Regressive Integrated Moving Average (ARIMA), Lyapunov exponent, exponential smoothing filter, dynamic threshold, and lastly, Rule-based classifier. We have evaluated the proposed RDAER model on the CICDDoS 2019 dataset, that achieved an accuracy level of 99.92% and a fast detection time of 20 s, outperforming existing methods.

Keywords Cloud computing, SDN, DDOS, Event correlation, DBSCAN clustering

Introduction

In the last decade, several researchers and developers have made a great effort to develop new computing technologies, creating a very complex digital environment where users can efficiently perform a range of jobs quickly and at a low cost. These technologies give consumers on-demand access to various services and resources. Cloud computing provides a digital platform for cloud users to access resources on demand based on a pay-per-use model [1]. Even the government and IT industries have shifted their focus to the cloud because it reduces the cost of infrastructure development and management. Virtualization is a critical technology in cloud computing as it gives service to a set of dynamically usable resources, such as storage, software, and processing power, over the internet [2]. Monitoring network traffic in a stable network structure presents a significant challenge for cloud providers. As a result, companies have turned to Software Defined Networks (SDN) as a preferred

*Correspondence:
Ganesh Reddy Karri
ganesh.reddy@vitap.ac.in
[1] School of Computer Science and Engineering, VIT-AP University, Amaravati, Andhra Pradesh, India

method for building networks over the past decade [3]. SDN simplifies the complexity of today's networks by converting physical network connections into logical network connections and providing centralized management of network services [4]. Cloud service providers can benefit from cost savings, intelligent global links, granular security, and reduced downtime with SDN [5]. SDN provides a software application plane for applications that offer practical solutions to essential network operations such as auto-scaling, intrusion detection, and network monitoring [6]. The development of SDN cloud networking, as depicted in Fig. 1, allows cloud service providers to host millions of virtual networks without relying on standard isolation methods such as VLAN. SDN represents a paradigm shift in network architecture. It decouples the control plane from the data plane, allowing network administrators to dynamically manage and control network traffic. In the context of cloud computing SDN enables the dynamic allocation of network resources to match the requirements of cloud applications and services. This means that as workloads in the cloud increase or decrease, the network can adapt accordingly, ensuring optimal performance. However, although SDN separates the control and data planes, it does not prevent network overload from traffic, resulting in DDoS attacks. Additionally, hackers can compromise the network's security by attacking several SDN components, including the controller, southbound and northbound interfaces, and the switch [7]. SDN-based cloud users face a significant issue with service disruptions caused by DDoS attacks.

A Distributed Denial-of-Service (DDoS) attack is probably the most well-known and dangerous threat to cloud computing. It can hurt both cloud providers and their customers. DDoS makes the help inaccessible to actual clients. Multiple nodes are compromised to generate the attack. The malicious user compromises multiple nodes to flood the target system with traffic [8]. A sample attack scenario is represented in Fig. 2. Recent estimates show DDoS attacks cause enormous financial losses for even the largest cloud providers, such as Amazon AWS EC2 and Rackspace [9]. DDoS attacks on servers and the infrastructure of the cloud [10]. Cybercriminals conducted around 5.4 million DDoS attacks in the first half of 2021, registering an up to 11%

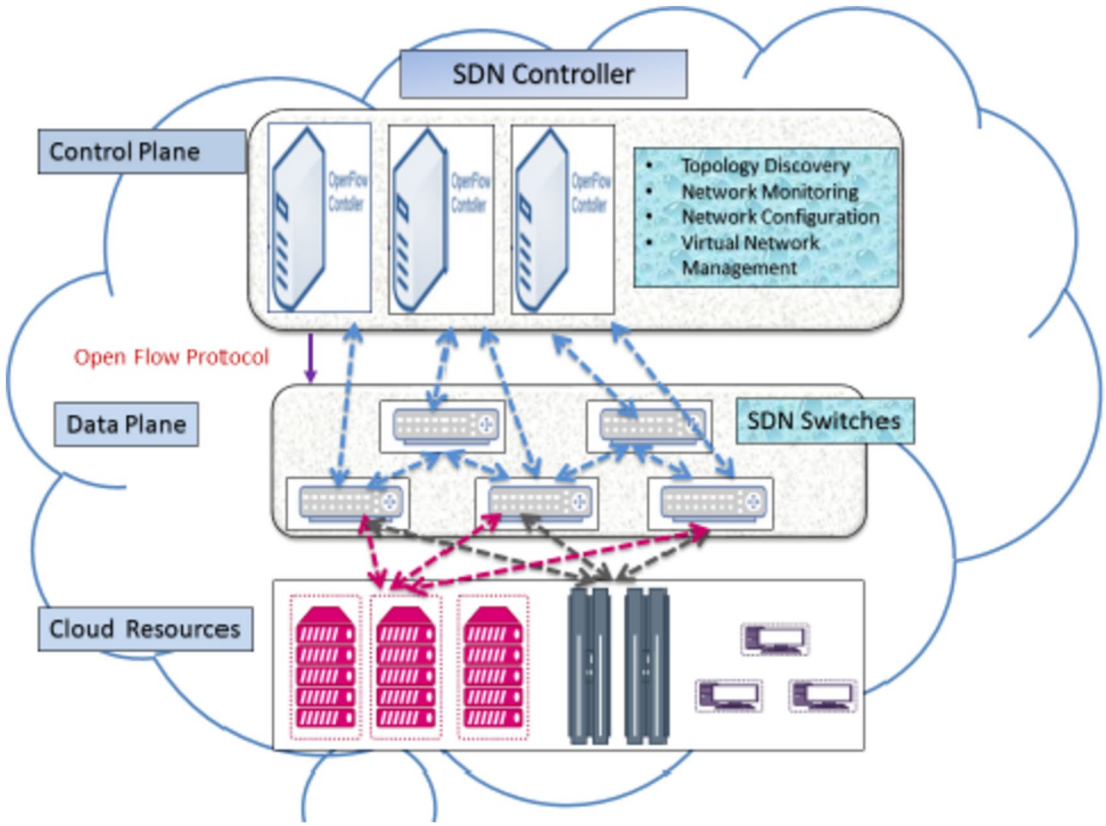

Fig. 1 SDN cloud networking

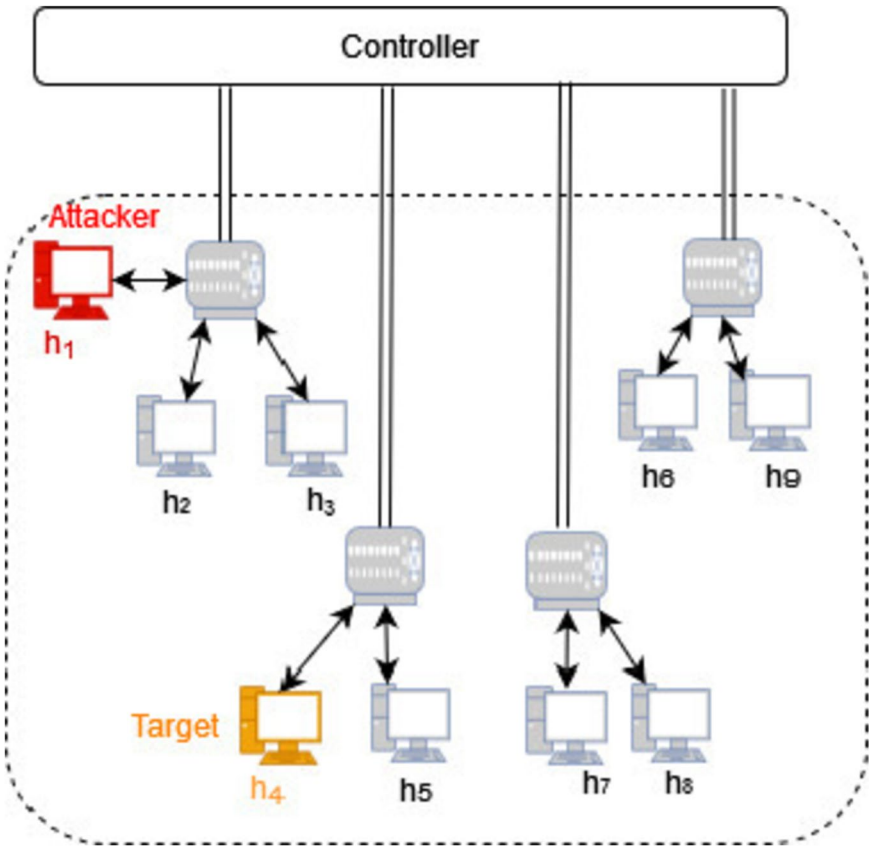

Fig. 2 Sample DDoS attack scenario

increase from the first half of 2020. An organization's ability to recognize and defend itself against DDoS is critical to its success.

Therefore, it is crucial to have a framework that can analyze network traffic and detect anomalies before any damage occurs [11]. An automated system that can classify network traffic and alert the controller is necessary [12, 13]. Although several DDoS defense systems are available, attackers continuously develop new attack patterns, making it challenging to detect anomalies early. While some existing strategies provide early detection, they have high false-positive rates [14]. Other approaches have high accuracy and detection times for DDoS but they can lead to resource outages and financial losses [15]. We introduce an innovative RDAER framework that seamlessly incorporates highly effective techniques within each category, including feature selection, traffic clustering, attack prediction, and traffic classification for SDN-based cloud networks. This integration aims to enhance the precision and timeliness of DDoS attack detection. To achieve this, we have incorporated the following techniques into our proposed framework:

- Perform data preprocessing by using multiple machine learning methods which convert raw traffic data into normalized data to improve the accuracy of predictions and effective resource utilization.
- For dynamic DDoS attack patterns, we use the Recursive Feature Elimination (RFE) method to select relevant features that accurately distinguish between benign and malicious data.
- To handle a huge volume of traffic data and dynamic DDoS patterns we use Density-Based Spatial Clustering of Applications with Noise (DBSCAN) to form the clusters based on time, which helps early detection of DDoS attacks.
- Furthermore, each cluster is analyzed using autoregressive integrated moving average **(**ARIMA), Lyapunov exponent, Exponential filters, and dynamic threshold to predict the chaotic behavior by calculating anomaly scores.

- Lastly, all the cluster scores are correlated using a rule-based event correlation classifier to determine whether traffic data is normal or a DDoS attack has occurred.
- We evaluate the effectiveness of the proposed RDAER framework by comparing it with the existing models in terms of accuracy and detection time. The results indicate that the RDAER framework outperforms accuracy and detection speed methods.

The commitments of the paper are as follows: Sect. 2 discusses related work. Section 3 deals with the proposed methodology. In Sect. 4, evaluation and experimental results have been explored, and the conclusion and future work are presented in Sect. 5.

Related work

In the past few years, scientists have presented several strategies for intrusion detection systems but barely any procedures for anomaly detection. These strategies face the challenge of creating a varied, flexible, and straightforward approach for abnormal behavior detection, given the complexity and speed of today's malicious behavior and the size of today's networks. The anomalies over a network can be detected using different intrusion detection techniques, namely mining, statistical, machine learning, and knowledge-based methods. Since early 2010, these techniques have been implemented individually to detect attacks leaving increased false positive rates [16, 17]. In 2015, as we have advanced to the next research phase, we can detect DDoS attacks by combining two intrusion detection approaches [18]. These studies have improved detection accuracy but at the cost of greater computational complexity and resource usage.

Later, a study was conducted on detecting and choosing suitable features which help decrease the detection time, to simplify matters [19]. Another model [20] employed several time series techniques for predicting DDoS attacks by forecasting the behavior of the traffic features at the time of attack using the anomaly scores. The model identified traffic as an attack or normal based on the scores. This paper [21] presents a new method for detecting DDoS attacks using Lattice Structural access rates; it is named S2RF2S for feature filtering, and it makes use of a Soft-Max Behavioral Based Ideal Neural Network (SxB2IN2) for classification. The work achieved an accuracy of 90% in detecting DDOS attacks. Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Random Forest (RF), and Long-Short Term Memory (LSTM) algorithms were used in another study [22] to emphasize feature selection for accurate and efficient identification. Compared to previous research, the RF classifier could achieve 99% DDoS detection accuracy with 11 features. In [23], a Spark tool was used to build a model for detecting DDoS attacks in SDN. In comparison to other algorithms, the Decision Tree (DT) has shown the best accuracy at 93.6%. Hence, DT was selected for real-time deployment. In another study [24], the widely used LSTM model was used to filter out suspect flows in distributed SDN-based edge computing. Extensive experiments on five different datasets with three common attack types demonstrated that the CoWatch framework achieved an accuracy of 93.30% in predicting and detecting DDoS attacks and their corresponding attack flows through a collaborative prediction algorithm.

Najafimehr et al. [25] used a hybrid model that combines both supervised and unsupervised learning methods. They utilized a clustering approach and many flow-based characteristics to distinguish attack traffic from regular data. The clusters are given by names using a classification method based on specific statistical measures. Phuc Trinch et al. [26] suggested an enhanced approach for detecting hacked SDN switches using a multivariate time series technique and Recurrent Neural Networks (RNNs) for classification. This work achieved an accuracy and detection rate of 96.99% and 98.51, respectively. Peng et al. [27] proposed an anomaly flow detection for SDN using the double *P*-value of the transductive confidence machines for the KNN algorithm. Using a sampling-based strategy, another study [28] proposed a scalable flow monitoring and classification solution for open flow. The classification method combines deep packet inspection and machine learning techniques. This paper [29] proposes use of machine learning to mitigate cloud-based DDoS attacks. The work covers gathering cloud module input data, reducing dimensionality, noise filtration, feature extraction, and ResNet-101-based Kernel Extreme Learning Machine (KELM) for classification.

In another study [30], the authors used agglomerative and K-means clustering with Principal Component Analysis (PCA) for feature extraction. A voting method classifies whether the data is normal or attacked. This method achieved a classification accuracy of 96.66%. Mosayeb et al. [31] proposed a 3-phase statistical model RAD to detect DDoS attacks by scoring users to classify them as attack or benign. The three parameters, drop, jitter, and delay, identify the potential attack behavior. RAD is tested using the CICDDoS2019 dataset and is compared to four other detection algorithms that achieved a precision of 80%

and a recall of 99%. Rajasree et al. [32] used a fuzzy bat clustering algorithm by grouping similar patterns and predicted the strange behavior by deviated anomaly score. The event correlation between the virtual machine instance supplied by the cloud service provider and the suspicious source list is established to identify the malicious source. This model has produced fewer false alarm rates when accurately determining the anomalies. Girish et al. [33] constructed a neural network using stacked and bidirectional LSTM models. Information collected from the open stack is used for testing the model. The information gathered includes ten characteristics and a classification label. Using the binary cross-entropy function as a loss function, the suggested model had a training set accuracy of 94.61% and a test set accuracy of 93.98%.

The analysis of existing studies highlights the absence of a comprehensive strategy that integrates clustering, time-series analysis, feature selection techniques, and event correlation for early DDoS attack detection in SDN cloud environments. Event correlation plays a pivotal role in identifying network patterns and anomalies across distributed networks. Additionally, there is a need to improve the DDoS detection accuracy. The novelty of our work lies in the combined application of clustering, time-series techniques, and event correlation, with a particular focus on the unique CICDDoS dataset. This dataset is essential for discovering new attack patterns that may not be present in older databases, emphasizing the importance of its utilization in uncovering unexplored threat scenarios. Table 1 compares existing works as well as their limitations.

Proposed system

The RDAER framework is designed for SDN SDN-based cloud environment. It comprises five modules: data preprocessing, feature selection, clustering, anomaly score prediction, and event correlation-based classification. The architecture of the RDAER is presented in Fig. 3. The SDN controller employs this approach and monitors each switch individually for DDoS attack traffic irregularities. The SDN agents at switches perform data preprocessing to convert the raw traffic from network flows and process it for normalized data. Using Recursive Feature Elimination (RFE), relevant features (Source IP address, Destination IP address, and timestamp) are chosen and then formed into clusters based on timestamp using the DBSCAN approach. Then, using time series techniques, each cluster is analyzed for any malicious traffic by releasing anomaly scores. Finally, the event correlation module correlates the final anomaly scores to classify the traffic sample as normal or DDoS. When a sample is abnormal, the framework sounds an alarm and activates the countermeasure section. Each module is briefly explained below.

Data preprocessing

In machine learning, data preprocessing is crucial in generating accurate and valuable results [34]. Data preprocessing improves data quality by handling missing or incomplete data, smoothing out noise, and addressing discrepancies. The following steps are involved in the preprocessing stage:

1. The correlated features get removed by selecting only one feature among many with a > 80% correla-

Table 1 Comparison of related works

Author	Model	Feature selection	Clustering	Classification	Event Correlation	Accuracy
Ramin Fadaei Fouladi [20]	ARIMA Time series	✓	×	✓	×	98.82
Karthick et al. [21]	S^2RF^2S and SxB^2IN^2	✓	×	✓	×	90%
Alubaidan et al. [22]	Random Forest	✓	×	✓	×	99%
Samaan et al. [23]	Chisquare and Random Forest	✓	×	✓	×	93.6%
Zhou et al. [24]	Optimal Threshold and LSTM	✓	×	✓	×	93.30%
Mohammad Najafimehr [25]	DBSCAN + ML	×	✓	✓	×	-198% more effective
Phuc Trinh Dinh [26]	Multivariate time series + RNN	×	×	✓	×	96.99
Peng [27]	KNN	×	×	✓	×	98.51
Saurez [28]	Deep packet inspection + ML	×	×	✓	×	98
Manjunath et.al [29]	Resnet-101 -KELM	×	×	✓	×	96%
Mohammed Misbah Uddin [30]	Agglomerative + PCA + ML	✓	✓	✓	×	96.66
Mosayeb [31]	RAD	×	×	✓	✓	99% recall
Raja sree [32]	Fuzzy bat clustering	×	✓	✓	✓	98.7
Our Paper	**RDAER**	✓	✓	✓	✓	**99.92%**

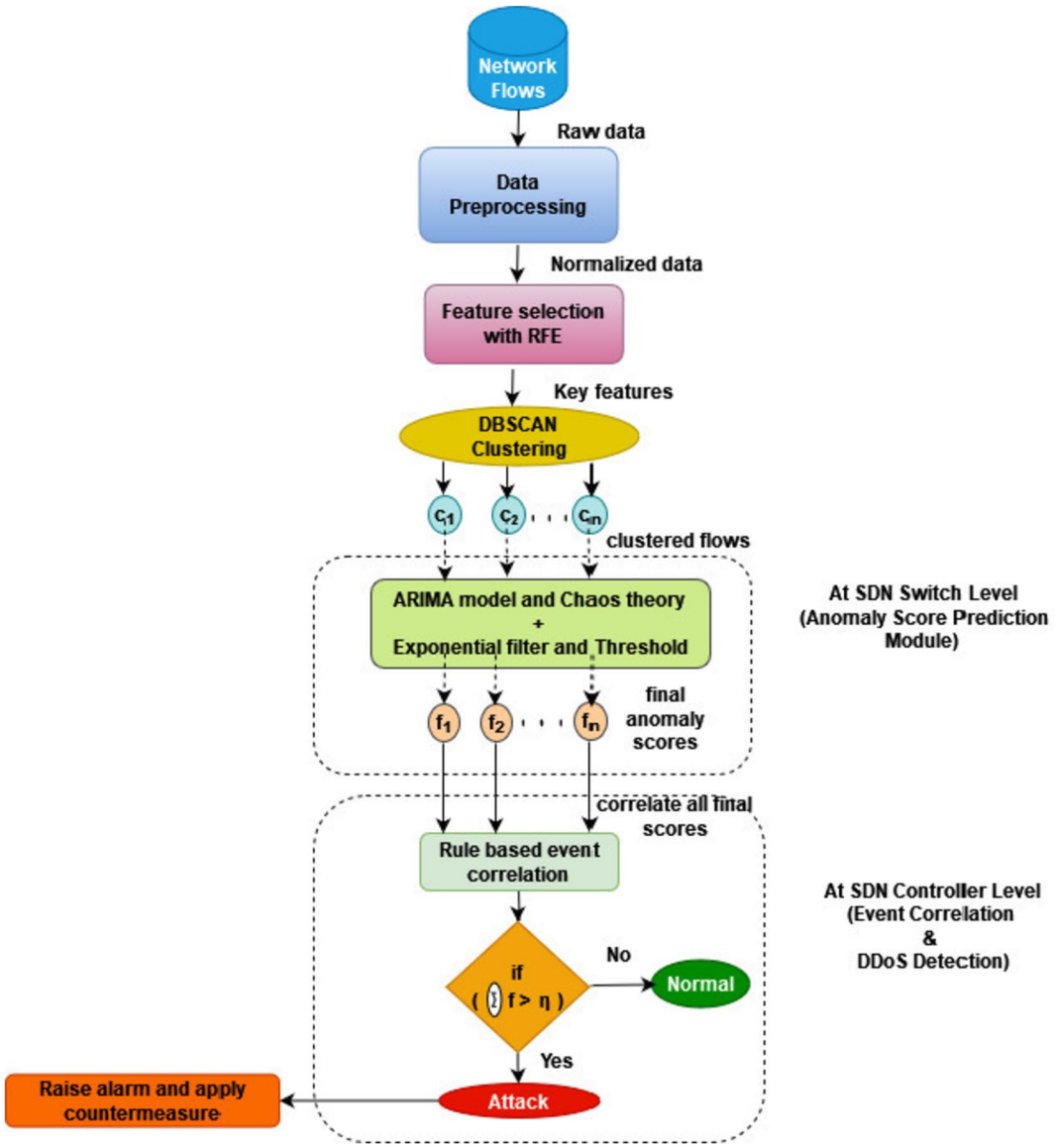

Fig. 3 Architecture of RDAER

tion. The Pearson correlation coefficient is employed, which gives a value between -1 and + 1 and can determine if two features have a linear relationship. The covariance of two features (p, q) is calculated using Eq. (1), where the cov (p, q) represents the covariance between two features. In contrast, σp and σq represent the standard deviation of p and q, respectively.

$$\rho(p,q) = \frac{cov(p,q)}{\sigma_p \sigma_q} = \frac{E[(p - E[p])(q - E[q])]}{\sigma_p \sigma_q} \quad (1)$$

2. To remove any incomplete data, we need to eliminate the rows that have missing values.
3. To replace infinite values with a maximum feasible value.
4. The data is normalized using the min-max scaling method, which involves applying the equation specified in Eq. (2). Here, z represents the value of a feature fe, while z' denotes the corresponding normalized feature value. The minimum and maximum values of the feature are denoted as $\min_{fe}$, $\max_{fe}$, respectively.

$$z^{'} = \frac{z - min_{fe}}{max_{fe} - min_{fe}} \quad (2)$$

5. The label encoding technique converts the categorical label column into a binary numerical array. Here DDoS is assigned the value of 1, while normal is assigned the value of 0.

Feature selection

Following the preprocessing phase, we perform feature selection on normalized data using the hybrid RFE approach described in the current paper. As DDoS attack patterns are in huge and high dimensional data overfitting, poor model interpretability, and longer calculation times are all possible consequences associated with it. The effectiveness of DDoS detection algorithms can be enhanced by using RFE to pick a subset of the most useful characteristics, thereby lowering the dimensionality. RFE constructs a model and selects the optimal or worst features based on their ranks, using a basic DT method as an estimator [35]. This method employs information entropy as a crucial measure for feature selection. It computes the information gain for each sample to divide it layer by layer until at least one sample type is separated [36]. The RFE with DT as an estimator provides rankings and importance scores for all features, as shown in Fig. 8. Based on the threshold, the source IP address and destination IP address have the highest ranking of all features and are selected for our work on DDoS detection.

During a DDoS attack, the number of flow entries with Unique Source IP Address (USIA) may grow due to fake and randomly produced IP addresses. In contrast, the number of Normalized Unique Destination IP Address (NUDIA) may not vary much compared to usual. Still, the normalized value of this statistic concerning the total number of packets in the flow table decreases. These two features are, therefore, independently used as time series in the attack detection procedure to identify potential cases of a DDoS attack. In addition to these two features, timestamps and class labels are also considered in this work, as they are associated with IPs. The feature list considered for our work is tabulated below in Table 2.

Table 2 Features selected

Sno	Feature name
1	Unique Source IP Address (USIA)
2	Normalized Unique Destination IP Address (NUDIA)
3	Timestamp
4	Label

Clustering

After feature selection, we cluster the selected features using the DBSCAN algorithm [25]. DBSCAN is a robust technique that can effectively control dynamic DDoS attack patterns, mitigating noise, and optimizing parameter settings. Though it may lead to increased energy consumption, it is still a viable option. Its versatility and adaptability make it an invaluable tool for studying and comprehending energy consumption patterns in dynamic time-series data, which enables the discovery of important clusters. This is made possible as DBSCAN possesses both of these qualities. The flow chart of DBSCAN is depicted in Fig. 4. Before clustering the data with DBSCAN, we first calculate the optimal value for the 'eps' parameter using (3), which

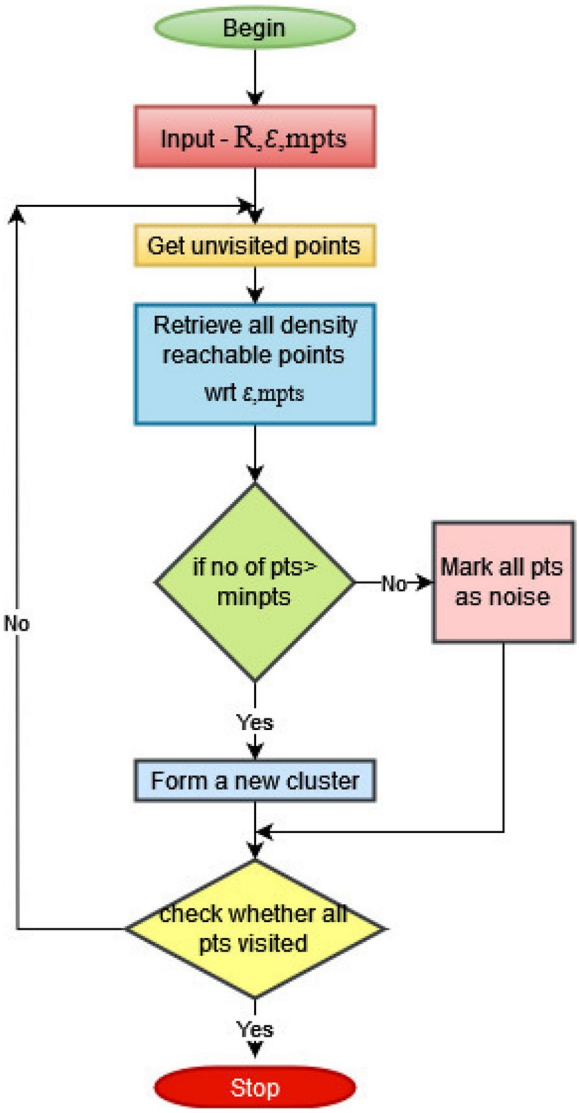

Fig. 4 Flow chart of DBSCAN clustering

retrieves all points that are densely reachable from point y, considering eps and minpts.

$$n_{eps} = \{y \varepsilon D / dist(x, y) < eps\} \quad (3)$$

If point y is a core point and Neps > minpts, we form a cluster and join the cluster's core point. We identify all border points with Neps, minpts, and all core points as neighbor's and mark the other points in D as noise points. We divide the features of USIA and NUDIA into different clusters based on timestamps. Then, each feature is processed separately using time series techniques in the next phase to calculate anomaly scores and detect DDoS attacks. However, each cluster is processed in parallel to accelerate the detection of DDoS attacks.

Anomaly score prediction module

In this phase, we separately analyze each cluster's USIA and NUDA features to determine their anomaly scores ($score_1$ and $score_2$) at time t. The anomaly prediction module is illustrated in Fig. 5. We apply the USIA feature to both the ARIMA and chaos theory methods to obtain $score_1$. For the NUDA feature, we pass it through exponential filters and dynamic threshold to get $score_2$. The notation used in the algorithms is tabulated in Table 3.

Processing of USIA feature

As said earlier USIA is passed as time series to ARIMA. ARIMA (p, d, q), is a three-tuple time-series forecasting statistical model [37], where p is the lag order, d denotes the number of times raw observations differenced, and q is the order of Moving Averages (MA) or lagged forecast errors as seen in Eq. 4.

$$Z^{'}(t) = c + \varphi_1 \times Z^{'}_{(t-1)} + \cdots + \varphi_p \times Z^{'}_{(t-p)} + \theta_1 \times \varepsilon_{(t-1)} + .. + \theta_q \times \varepsilon_{(t-q)} + \varepsilon_t \quad (4)$$

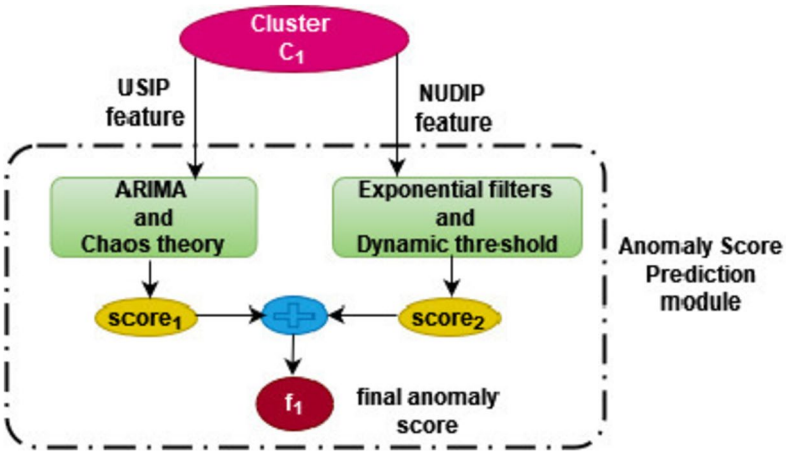

Fig. 5 Anomaly score prediction module

Table 3 Notation used in the work

Notation	Meaning	Notation	Meaning
$Model_1$	*Training flag for USIA feature*	$a_{1\ and}\ a_2$	*a_1 as 0.1 and a_2 as 0.8*
$Model_2$	*Training flag for NUDIA feature*	$f1(a_1)$ and $f2(a_2)$	*Exponential filters a_1 as 0.1 and a_2 as 0.8*
$Z \leftarrow \{z_1, z_2, \ldots, z_t\}$	*Timeseries of unique Source IP*	Ad_f	*Time series of distance between f1, f2*
$Y = \{y_1, y_2, y_3, \ldots y_t\}$	*Time series of normalized Destination IP*	M	*Timeseries for rolling median of Ad_f*
$ARIMA\ (p, d, q)$	*p is the lag order, d is the number of times raw observations differenced, and q is the order of Moving Averages (MA)*	Id_t	*The minimum distance between items in M*
λ_t	*Lyapunov exponent at time t*	η	*Threshold value*
p_t	*ARIMA model Prediction error at time t*	q	*The threshold is defined by the standard deviation number, σ_{Id}*
$Score_1$ and $score_2$	*Anomaly prediction scores of Z and Y*	σ	*Standard deviation of Id*
a,	*Exponential filter's Smoothing constant*	μ	*Mean of the least distance*

In Eq. (4), the term $Z^{'}(t)$ is a time series, ϕ_1 and $theta_1$ are the first Auto Regression (AR) and MA terms, and p and q are the order of AR and MA terms, respectively, and finally, ε_t is the error. ARIMA captures the trends and seasonality of the network traffic and allows checking for any spikes and fluctuations in the traffic. Spikes relate to abnormal traffic. To find an inaccuracy in prediction error, we use the Lyapunov exponent, as depicted in Eq. (5) below:

$$\lambda t = 1/tln(|pt/p0|) \tag{5}$$

where p_0, p_t and λ_t represent the first prediction error, the t_{th} prediction error, and the Lyapunov exponent at the t_{th} instance, respectively. Positive exponents imply DDoS traffic, while negative exponents indicate regular traffic [38].

According to algorithm 1, the ARIMA model estimates the attack trend of the sample set Z, where z_t is an exponential function of time t. To construct the ARIMA model, set $Model_1$ to be true. For training the model, n samples of source IPs are stored in Z. If Z is non-stationary, apply differencing $d > 1$ to achieve a stationary time series. Differentiating the time series makes Z suitable for stationary time series analysis and modeling. The Box-Cox transformation stabilizes the variance of a time series variable y. The Box-Cox transformation is a mathematical technique that adjusts the data distribution to make it more suitable for analysis and modeling. It is also possible to use either Akaike's information criterion, the corrected version of this criterion (AICc), or the Bayesian Information Criterion (BIC) to select the order model [39]. By minimizing these criteria, the best model was selected. Figure 6 is the differencing graph that shows in which order of differencing data is stationary and Fig. 7 specifies whether there are any outliers for the order selected. The peaks in the density plot specify the anomalies. The ARIMA model generates the standard feature pattern, but no attack instances should occur during the model generation. After model generation, the $Model_1$ flag is set to FALSE, and the training phase is completed. In the testing phase normal model estimates the value, $\hat{z}_t$, for each subsequent incoming traffic sample, z_t. If any attack traffic is coming the model predicts an abnormal behavior by generating the spikes. The prediction error's chaos calculates an anomaly score [40]. The prediction error p_t is determined using Eq. (6)

$$pt = \left|(z_t - \widehat{z}_t)\right| \tag{6}$$

To assign an anomaly score for different outcomes of prediction errors, the Lyapunov exponent (λ) is used. According to Eq. (5), a positive value of λ indicates attack traffic (score $_1 = 1$), while a negative value suggests normal traffic (score $_1 = 0$).

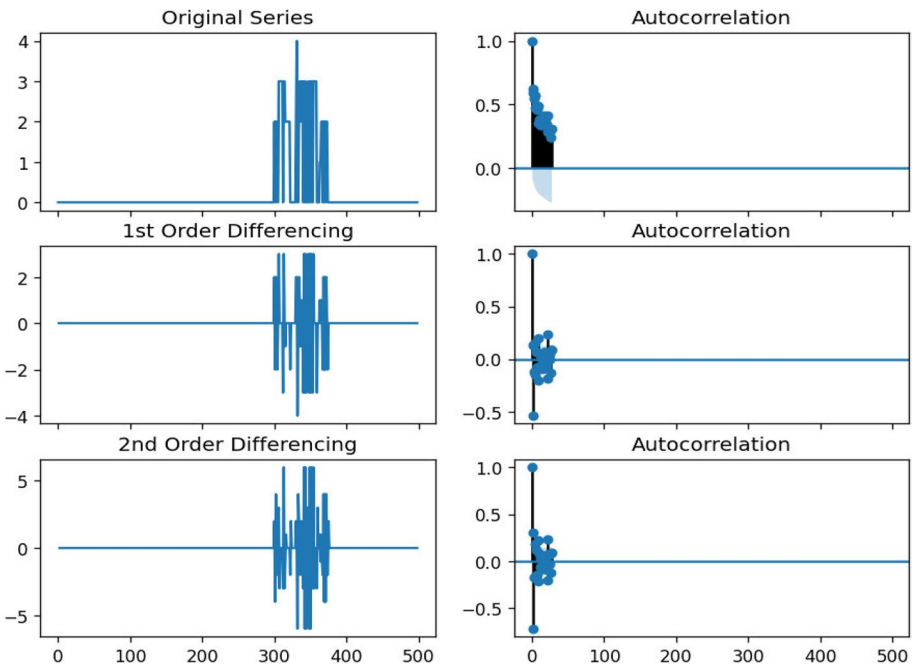

Fig. 6 Differencing graphs for selecting the ARIMA model

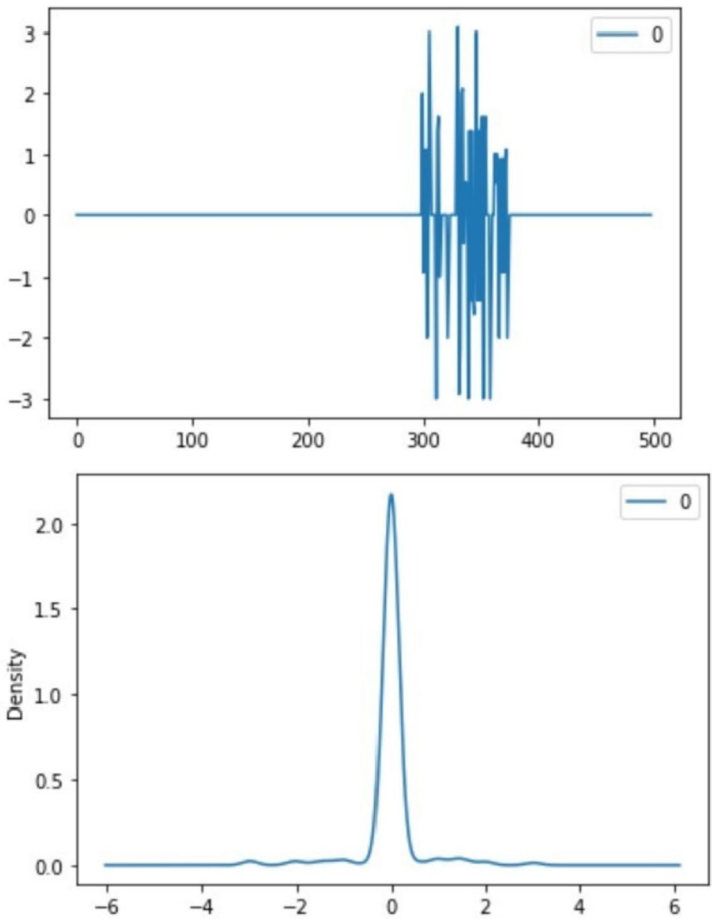

Fig. 7 Line plot and density plot of residuals

Algorithm 1 Prediction of Anomaly score$_1$

INPUT: *USIA feature set* $\{z_1, z_2, \ldots, zt\}$
OUTPUT: *prediction of* $score_1$

// Training phase
Set $Model_1 = TRUE$
$z_{count} = 0$;
$Z \leftarrow \{z_1, z_2, \ldots, z_t\}$ *// USIA feature*
At a time instance t
  z_{count} ++
  Train ARIMA $_{(p,\ d,\ q)\ for}$ Z
  After completion of the training phase set $Model_1 = FALSE$
// Testing phase
  $\hat{z}_t \leftarrow ARIMA\ (p, d, q)(\{z_{t-p-d}, \ldots, z_{t-1}\})$
  $p_t \leftarrow mod\ (z_t - \hat{z}_t)$ *// Estimate prediction error*
  calculate Lyapunov exponent λ_t
  if $\lambda_t \leq 0$, *then*
    Normal traffic (score = 0)
  else
    DDoS traffic (score = 1)
  end if
end

Processing of NUDA feature:
Here exponential smoothing forecasting model is used, which gives weights to the earlier and new observations for forecasting. New observations have higher weight than earlier observations based on the smoothing constant α, which makes things look smoother. The value of α is between 0 and 1 as per Eq. (7) where E_i signifies the smoothed data and x denotes the original data.

$$E_t = \alpha.E_{t-1} + (1 - \alpha).Z_t \tag{7}$$

In algorithm 2, the n samples of the NUDIA feature are used and stored in Y as a time series to generate the model. Y is estimated by two exponential filters, f1, and f2, with their exponential constants α_1 as 0.1 and α_2 as 0.8 and their absolute difference stored in Ad_f. The rolling median generates a median time-series, M. The least distance between each case and the remaining samples is determined and stored in a set, ld. The mean μ_{ld} and standard deviation σ_{ld} are computed. $Model_2$ is set to False once the above-stated values

are determined. Now for each y_t feature of upcoming traffic, the process mentioned above is repeated, and the least distance ld_t is calculated. If it is less than the threshold value $\eta = \mu_{ld} + q^* \sigma_{ld,}$ the traffic instance is considered normal ($score_2 = 0$); otherwise, it is abnormal ($score_2 = 1$).

Algorithm 2 Prediction of anomaly $score_2$ using smoothing filters

INPUT: *NUDIA feature set Y*
OUTPUT: *Anomaly* $score_2$
$Model_2 \leftarrow True$
$y_{cnt} \leftarrow 0$
$Y = \{y_1, y_2, y_3, \ldots y_t\}$
At time t
  y_{cnt} ++
  If $y_{cnt} \geq 1$
    Initialize and assign two filters f1, f2 with Y[1]
    From y[2:] find
    $f1_i = \alpha_1 . f1_{i-1} + (1 - \alpha_1) . y_i$
    $f2_i = \alpha_2 . f1_{i-1} + (1 - \alpha_2) . yi$
    Calculate the Absolute difference AD for the output of two filters,
    Followed by median time series $M = median\ (AD_i, \ldots . AD_{i+w})$
   Find the least distance ld and its mean and SD
   calculate threshold $\eta = \mu_m + q * \sigma_m$
  end if
  find $f1_t, f2_t, AD_t, m_t, ld_t$ \\ *testing phase*
  if $ld_t < \eta$ *then*
      Normal traffic ($score_2 = 0$)
      else
      Attack traffic ($score_2 = 1$)
  end if
end

The anomaly score prediction module collects score1 and score2 from the above methods and performs ANDing operation to obtain the final anomaly score f. All the collected final scores from each cluster are fed to next the module for DDoS detection.

Event correlation-based DDoS detection

The utilization of a rule-based method for network event correlation is very important in the identification of DDoS assaults in network settings. The methodology encompasses the gathering of data from multiple network nodes, performing preprocessing to assure uniformity, and afterward using predetermined correlation rules specifically designed to detect patterns indicative of DDoS attacks. These rules look at the spatial, temporal, and rate-based parts of network traffic, keeping an eye out for sudden traffic spikes, strange protocols, or high resource usage. A rule triggers an alert describing the nature and severity of potential DDoS activity. This alert initiates further research to protect network resources. These rules are updated based on real-world incidents and emerging threats to provide proactive and adaptive DDoS detection and prevention. Due to the event correlation, detecting the attack traffic too early with reasonable accuracy is possible, which may reduce the economic loss and huge damage to resources in the cloud network.

According to algorithm 3, calculate the threshold η for classifying abnormal and normal traffic. The rule-based classifier function calculates the sum of final anomaly scores for all clusters and checks if it exceeds the threshold to determine the traffic type. The main loop simulates the continuous processing of incoming traffic samples. Inside the loop, anomaly scores are calculated for each cluster and collected in the cluster scores list. If abnormal traffic is detected, an alarm is raised by the controller. The corresponding IP address and its corresponding switch are added to the discarded list. A defense mechanism can stop a DDoS attack but also stop any packets sent to a victim's IP address. As a result, immediately after the attack, the controller must change the activity of the flow entries.

Algorithm 3 Event correlation and DDoS detection

Input: *Final Anomaly Scores from different clusters*

```
# Define a function for event correlation
    def Rule based classifier (cluster_scores):
# Calculate the sum of anomaly scores for all clusters
    total_score = sum(cluster_scores)
# Check if the total_score is greater than the threshold
    if total_score > threshold:
    return "Abnormal Traffic"
    else:
    return "Normal Traffic"
# Main loop to process incoming traffic samples
    while True:
    incoming_traffic = receive_traffic_sample()  # Receive a new traffic sample
# Process the traffic sample and obtain anomaly scores for each cluster
    cluster scores = []
    for cluster in clusters:
    score1, score2 = calculate_anomaly_scores(incoming_traffic, cluster)
    final_score = score1 AND score2  # Perform AND operation as described
    cluster_scores.append (final_score)
    calculate threshold η based on mean and standard deviation of final scores.
# Perform event correlation to determine traffic type
    traffic_type = Rule based classifier(cluster_scores)
```

RDAER working model in cloud environment

The RDAER framework showcases strong adaptability to the scalability challenges within expansive SDN-based cloud networks. Acknowledging the increasing presence of multiple controllers, switches, and routers as the network expands, the framework is tailored to accommodate the network's growth. Its design leverages the hierarchical structure of SDN-based cloud networks, enabling robust event correlation at different network levels. Event correlation ensures early and accurate detection of malicious traffic, effectively reducing false positives. By proactively addressing scalability concerns and adapting to complex, multi-tiered network infrastructures, the framework demonstrates its capacity to maintain efficiency and accuracy even in the face of substantial network expansion.

The RDAER framework suggests a decrease in computational resources even when network size grows. This reduction is due to the utilization of only two features for analysis in contrast to more complex models that may employ a higher number of features. As the framework is scalable, the computational load is reduced leading to less resource utilization.

In one instance, in the scenario of a server inundated with massive traffic, the legitimate connection attempts are erroneously flagged as malicious, leading to false positives. Another situation involves failure of proper filtering at the switch level, incorrectly categorizing malicious traffic as benign, that results in false negatives. Our framework tackles these false alarms through a multi-layered strategy. By combining feature selection, traffic clustering, anomaly prediction, and event correlation, the framework enhances the precision of attack identification while reducing false alarms. This multifaceted approach, applied across various network levels, indicates a comprehensive strategy aimed at diminishing false negatives by capturing diverse patterns of DDoS attacks. Additionally, the focus on consolidating data through event correlation highlights the need for evolving a method to minimize false positives by establishing a contextual understanding of network behavior.

Experimental evaluation

Python programming, the Scikit-learn library is used to evaluate the proposed RDAER system. To assess the proficiency and viability of our strategy in identifying malicious traffic, we executed the proposed approach on a best-in-class dataset.

Performance metrics

This paper addresses the issue of separating malicious traffic from legitimate traffic. When evaluating the performance of a detection model, one should consider taking several metrics into account as given below:

$$Accuracy(Acc) = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$Precision\ (\mathrm{Pr}) = \frac{TP}{TP + FP} \tag{9}$$

$$Recall(Re) = \frac{TP}{TP + FN} \tag{10}$$

$$F1score = \frac{2TP}{2TP + FP + FN} \tag{11}$$

where TP (True Positive) denotes the number of malicious samples the algorithm has found; TN (True Negative) represents the number of benign samples to the normal ones; FP (False Positive) represents normal samples that are mistaken for malicious ones; FN (False Negative) denotes attack samples identified by false negatives.

The area under the ROC curve (AUC) is a measure of efficiency that considers all possible classification levels. DDoS attacks can be detected using a model with a high AUC value.

Dataset

The dataset we used to evaluate the proposed RDAER approach is CICDDoS2019 which the Canadian Institute gave for Cybersecurity [41]. This dataset is the naive dataset with more modern attacking methods. Reflection and exploitation attacks are the most common types of attacks in the dataset. These attacks mask the intruder's identity by sending packets to servers from the target IP address, causing the target victim's bandwidth to become overburdened with response packets. The dataset is composed of 88 features. It provides 12 types of DDoS attacks, namely NTP, DNS, LDAP, NetBIOS, UDP, UDP-Lag, SSDP, SYN, TFTP, SNMP, MSSQL, and Web DDoS [42]. Considering the experimental configuration's network infrastructure, the interval was t = 1 min. The following results are based solely on examining the dataset CICDDoS 2019. Overall, 500 traffic samples were employed for our experiments. The first 200 samples train the network's normal behavior, while the remaining 300 test it.

Detection performance

During model training, 86 features are given to the RFE method which has given ranking to all the features. The feature importance graph is shown below in Fig. 8. The graph shows that the source address and destination address secure the highest ranking which is above 0.08 than all features. The standout feature of RDAER is its focus on feature reduction. By utilizing only two key features for DDoS detection, it effectively minimizes resource utilization and detection time. Following the extraction of features, the number of clusters for the proposed RDAER was three, with the optimal values of eps and minpts as 0.08 and 7, respectively. Thus, we estimate the DBSCAN for homogeneity, completeness, the adjusted Rand index, mutual information, and the silhouette coefficient. Figure 9 shows the DBSCAN result.

The USIA feature of a traffic sample is processed for score_1 using ARIMA and chaos theory. The trained model is used to forecast the following value for $\hat{z}_t$. Figure 10 depicts the original and predicted values for the $\hat{z}_t$. The projected value differs from the actual value during the attack since the ARIMA model was trained

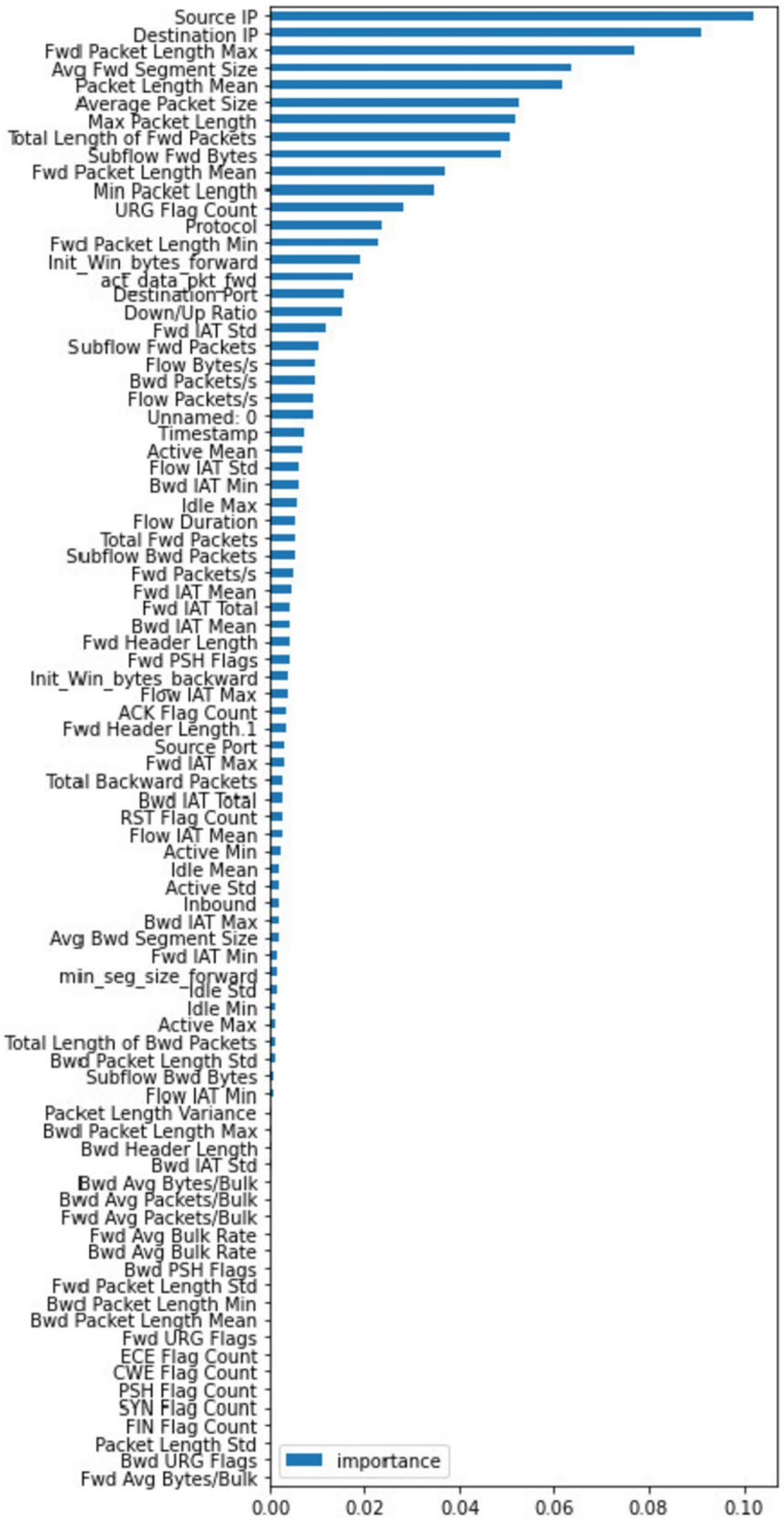

Fig. 8 Feature importance graph

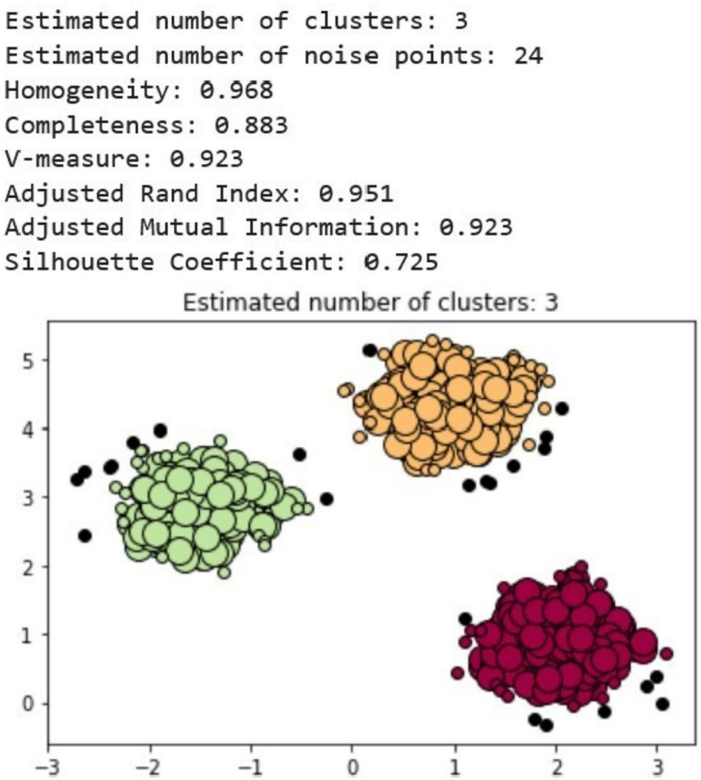

Fig. 9 DBSCAN clustering

using regular traffic data. The error's chaotic behavior distinguishes attack samples from the usual traffic flow. Hence, the prediction error is estimated using the Lyapunov exponent. From Fig. 11, the negative Lyapunov exponent value determines the regular traffic, whereas the positive value determines attack traffic. We map negative and positive Lyapunov values to scores 0 and 1, respectively.

The exponential filter and threshold method uses the same traffic sample to calculate the anomaly score2 for the NUDA feature. The two exponential filters with different α values calculate the anomaly $score_2$. The output

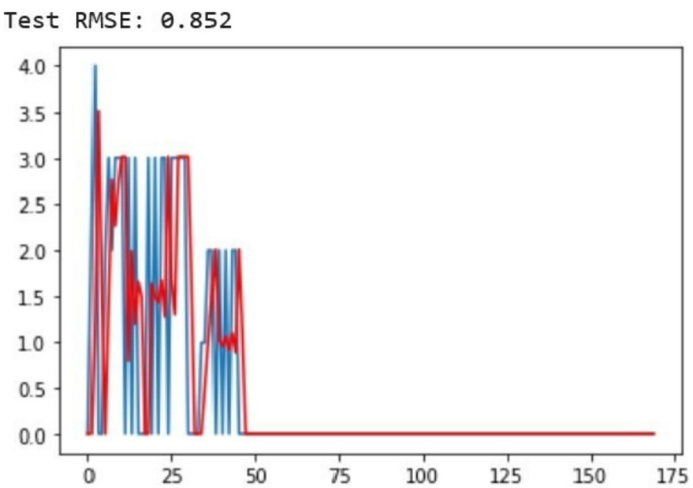

Fig. 10 ARIMA prediction

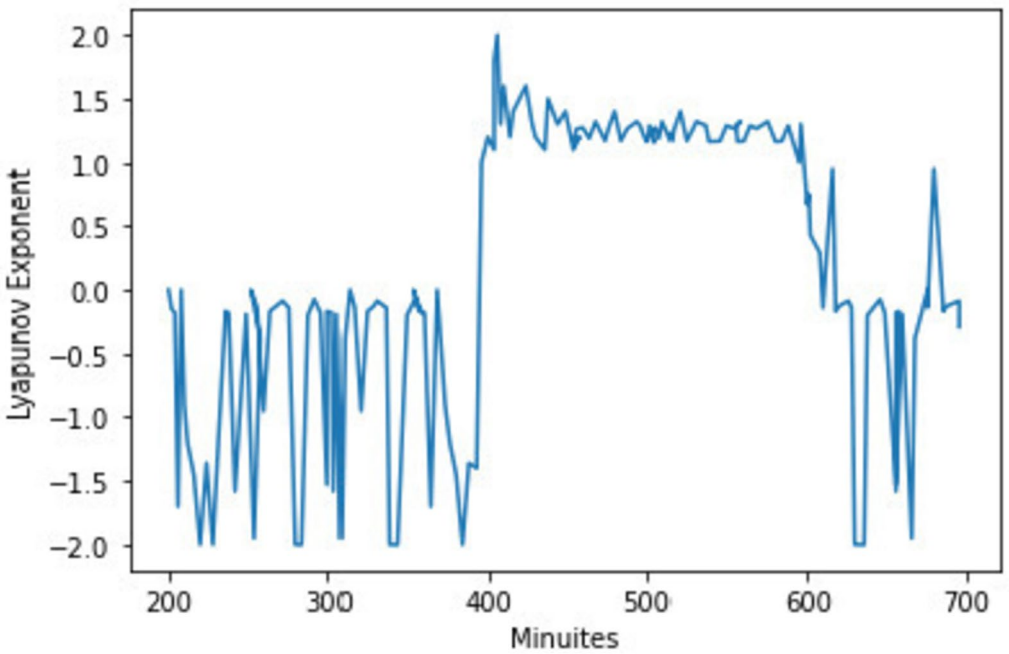

Fig. 11 Lyapunov exponent for ARIMA

of the filters and their difference in estimating the new feature is presented in Figs. 12 and 13. When a rolling median is applied to the difference with a window size of w = 5, a median is produced, which is used as the new feature to differentiate between regular traffic and attack traffic. Data's mean and standard deviation generate the threshold values. The median is larger during an attack than during a normal one as represented in Fig. 14. The median is high between 400 and 600 min, which indicates an attack is occurring throughout this period. To get the final anomaly score for one cluster, an ANDing of $score_1$ and $score_2$ was performed. The rule-based method correlates all the final anomaly scores obtained from each cluster and then determines whether or not the specimen is abnormal. As the clusters are correlated, the detection of attack traffic will be speedy. The result of the detection method was depicted in Fig. 15.

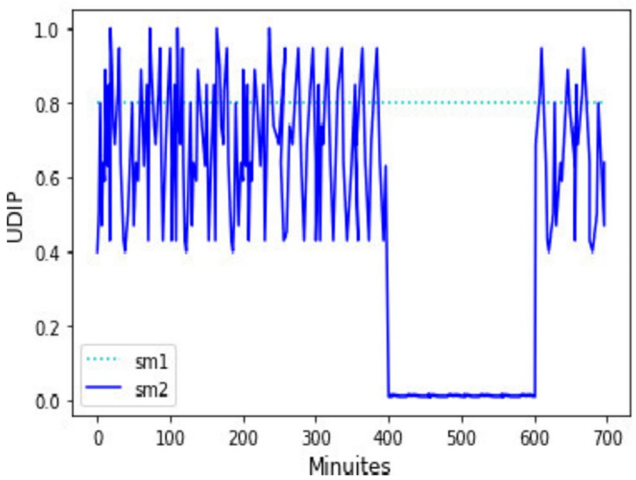

Fig. 12 Smoothing exponent

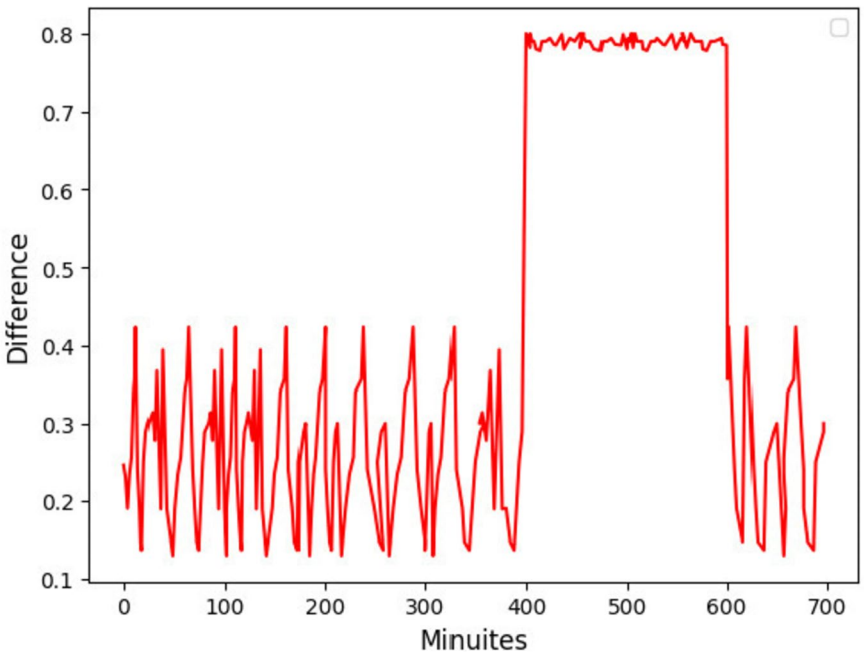

Fig. 13 Absolute difference

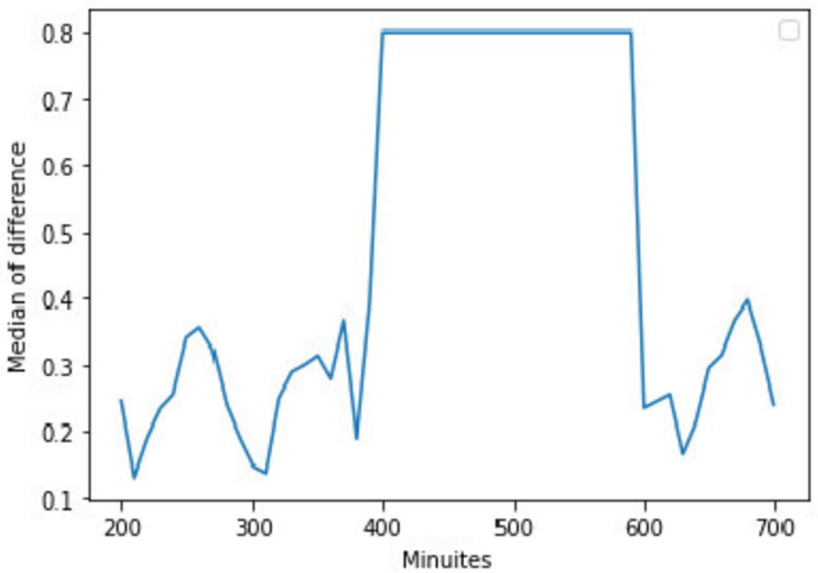

Fig. 14 Median of difference

Comparison with state of the art detection methods

This section presents a comparative analysis of the proposed model with recent models against different techniques used in the RDAER model. We evaluate various aspects of the models and provide insights into their performance and effectiveness.

Table 4 shows comparison between the DBSCAN and other clustering techniques. Compared to alternative clustering techniques, it is obvious that the DBSCAN method performed the traffic data analysis more efficiently and with a very high degree of accuracy. The DBSCAN algorithm removes noise when evaluating

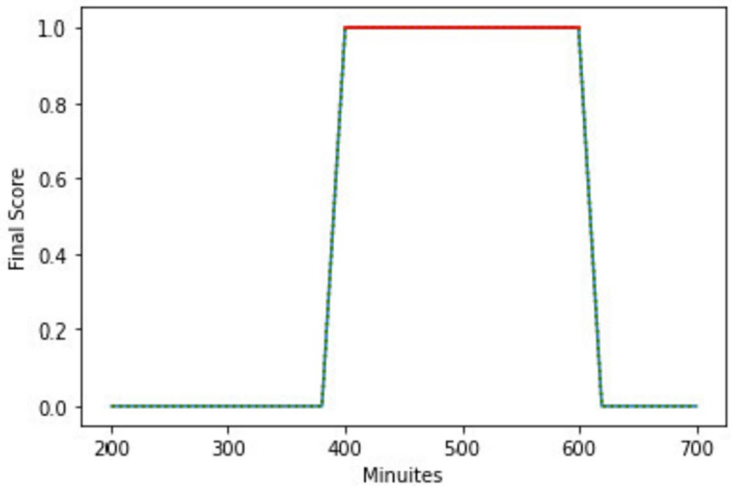

Fig. 15 Final score

Table 4 Comparative analysis of RDAER vs. other clustering techniques

Authors	Clustering algorithm	Acc	Limitations
M Aamir et al. [30]	AC and (PCA + Means)	96.66	Detection is done on the CICIDS2017 dataset
C. Ates et al. [10]	fuzzy clustering	98.7	Cannot identify novel attacks
Y. Gu et al. [43]	Supervised K-means + HFS	96.5	The novel dataset is not used and accuracy is also much less
Jasim et al. [44]	K-means	79.60	Accuracy is very low
RDAER Proposed	**DBSCAN**	**99.92**	**Training and testing data can be further tuned to improve the accuracy**

unlabeled data to prevent false positives. It is also noticed that by dividing the data into clusters based on timestamp and by processing all the clusters at the same time, you can get faster response times, which is important for early DDoS attack detection and reduced training time of the model. In RDAER, a notable advantage lies in the fact that DBSCAN operates with just three features (USIA, NUDIA, and timestamp), resulting in minimized resource usage and expedited detection processes.

Table 5 shows the comparison of the proposed RDAER with other time series techniques. These techniques establish baseline patterns of normal network behavior. When incoming data significantly deviates from these established patterns, indicating unusual spikes or irregularities, it may signal a potential DDoS attack. With the utilization of time series techniques for the traffic features (USIA and NUDIA), they indicated unusual spikes at a particular time period between 400 to 600 min range. This pattern showcases the malicious behavior associated with the network traffic and which could be a DDoS pattern. Utilizing two distinct techniques for analyzing the two features at a specific timestamp accelerates the

Table 5 Comparative analysis of RDAER and other time series techniques

Authors	Time series Classifiers	ACC	Limitation
Xinqian Liu et al. [45]	Dynamic Threshold	98	Accuracy is low
Jisa David et al. [46]	GARCH and ARMA	99.6	The detection time is too high
Maheshwari et al. [47]	MapReduce and Time series	NA	The detection time is 5 min which is too high
Alghawli et al. [48]	Entropy, signature analysis	97	The detection time is more because of packet learning
RDAER Proposed	**ARIMA + Exponential filters + Rule-based correlation**	**99.92**	**Training and testing data can be further tuned to improve the accuracy**

prediction process, resulting in swift detection—an area where previous models faced limitations. One more crucial aspect of RDAER is its capability to predict anomalies at the switch level, facilitating early detection. This ability significantly mitigates potential losses caused by DDOS attacks. From, Table 5 it is clear that the combination of ARIMA and exponential filters achieved good accuracy and less detection time in detecting the DDoS attack. Other time series methods are less accurate than RDAER.

The RDAER model is also compared with the latest DDoS detection techniques against accuracy, precision, recall, and f1-score, as shown in Table 6. Another study [20] proposed a time series model and showed that the model they developed achieved an accuracy of 98.82% in detecting DDoS attacks. The author of the paper [49] proposed an extreme learning algorithm that detects DDoS attacks with an accuracy of 99.18% with the NSL-KDD dataset and 95.11% with the ISCX dataset. Another learning-based K-means and optimal fuzzy system model [50] achieved an accuracy of 96.54% in detecting intrusions. The RNN-based model [51] achieved an accuracy of 94.12% and a precision of 98.18% in detecting the attack. The paper [52] presented a model based on cognitive mechanisms termed artificial immune systems for detecting DDoS attacks in the cloud environment with the accuracy and precision levels of 96.56% and 95%, respectively. In [53], a deep neural network with an auto-encoder approach is proposed for detecting DDoS attacks and has achieved a good accuracy of 98.43%. Table 6 shows that while some frameworks utilize a single method and others employ combined techniques, none have integrated an event correlation technique in a distributed network. The standout feature of RDAER is its event correlation capability at the controller level. Utilizing a rule-based classifier, the RDAER framework conducts event correlation across various network anomalies, resulting in remarkably high accuracy and reduced detection time when identifying attack traffic. The graphical representation of comparison of RDAER with earlier models is depicted in Fig. 16.

Table 7 distinguishes the performance metrics between the RDAER and up-to-date models based on the CICDDOS 2019 dataset. The given table presented year-wise models implemented on the DDoS2019 dataset. The proposed work outperformed existing models in terms of accuracy, and utilized only two features compared to other models, thereby reducing resource utilization and computational costs. This more efficient approach allowed the model to detect attacks early with high accuracy. The outcomes indicate that this defense technique is successful in preventing DDoS attacks. In the event of an attack, the controller promptly modifies flow table entry rules upon switch detection, effectively isolating harmful traffic and ensuring network security.

Table 8, and Fig. 17 presents the comparison of detection time between RDAER and other techniques. Authors in another study [59] used the IFS method for detecting DDoS attacks with a detection time of less than 300 s. The Logistic Regression technique [60] achieved a success rate of 99.8% and a detection time of 788 s. In [61], the author detected the attack in 40.78 s using a hybrid machine-learning technique. Another author in [41] witnessed the DDoS attack in minutes. Other models postulated previously [62] and [15], show a detection time of 320 s and 24 s, respectively, for predicting DDoS attacks. Compared to earlier models, the proposed model showed promising results with 99.92% accuracy and less detection time (by 20 s) using feature selection, traffic clustering, time series and event correlation techniques. Several factors contribute to the reduced detection time. Firstly, the use of only two specific features, their clustering, and parallel processing enable the initial prediction of anomalies at the switch level. Additionally, the consolidation of these findings through event correlation at the controller level significantly diminishes the time required for identifying DDoS patterns. Figure 18 shows the ROC curve, and the value of AUC achieved is 1.0.

Table 6 Comparative analysis of RDAER vs. existing works

Model	Acc	Pr	Re	F1 score	AUC	Limitations
ARIMA [20]	98.82	98.46	0.8	0.98	0.99	It was less accurate. It only employs one hidden layer
V-ELM [49]	99.18	NA	99.5	NA	NA	There is potential to improve accuracy
LKM-OFS [50]	96.54	84.89	89.92	85.66	NA	The dataset utilized is KDD1999 and might not detect new attacks
RNN [51]	94.12	98.19	95.13	93.56	NA	When used with novel DDoS datasets, it might still show lower accuracy
AIS-Based [52]	96.56	95	96	NA	NA	Although detection time is shorter, accuracy is also lower
AE + DNN [53]	98.43	99.22	97.12	98.57	NA	It is a complex analysis, and when applied to fresh attacks, detection times may lengthen and accuracy may decline
RDAER	99.92	99.9	99.9	99.9	100	**Training and testing data can be further tuned to improve the accuracy**

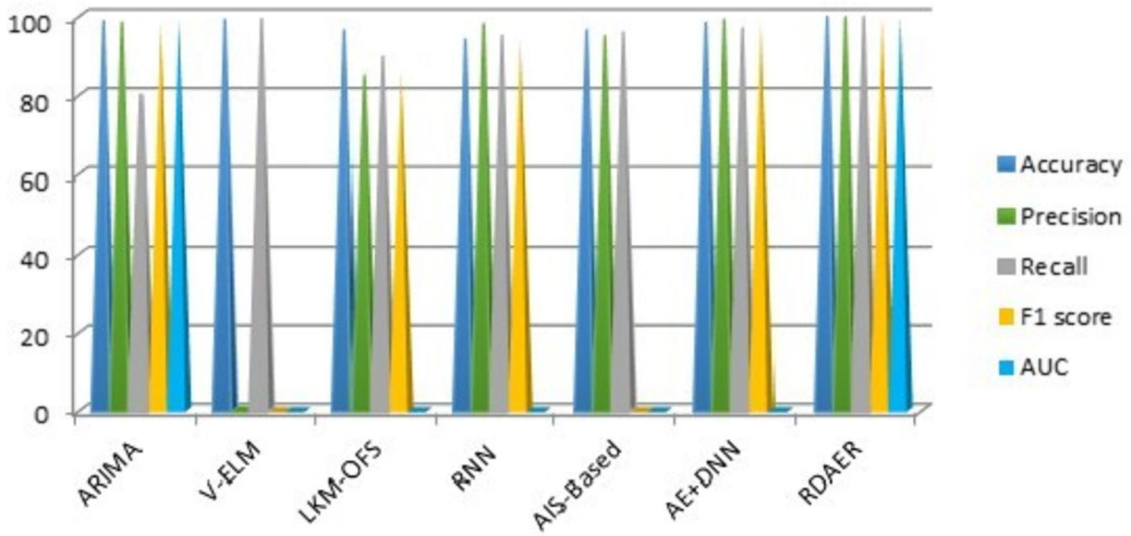

Fig. 16 Comparison of RDAER with earlier models

Table 7 Comparative results of RDAER vs latest schemes based on the CICDDoS2019 dataset

Models	Year	Accuracy	Precision	Recall	F1score
Bagging [54]	2020	96.9	96.9	96.4	96.2
DT [55]	2020	93.83	94.56	NA	93.21
RF [55]	2020	95.19	95.1	NA	94.47
KNN [55]	2020	87.3	85.78	NA	87.12
Decision [56]	2021	97	99	97	97.8
Stacking [57]	2021	97.3	NA	96	NA
BPNN [58]	2022	97.7	4.4	99.9	97.13
RDAER	**2023**	**99.92**	**99.9**	**99.9**	**99.9**

Table 8 Comparison of detection time between proposed and existing methods

Model	Detection Time
IFS Method [59]	< 300 s
DT [60]	1043 s
GB model [61]	40.78 s
ID3 [41]	few min
MTSF Model [62]	320 s
TPANGND [15]	24 s
RDAER (Proposed system)	**20 s**

Case study

In a cloud-based organization, the network infrastructure is tiered to accommodate various user levels, offering Software as a Service (SaaS) to its clients. This setup includes multiple routers and switches managed centrally by an SDN controller. As data moves through these switches, the flow tables store traffic patterns, coordinated by the controller. At this juncture, our RDAER system steps in to extract raw data and preprocess it, ensuring the removal of missing or null values. Subsequently, the preprocessed data undergo feature selection using the RFE algorithm, focusing on the most pertinent attributes. These selected features are then grouped based on timestamps using the DBSCAN algorithm and are scrutinized within clusters for any irregularities or signs indicative of a potential attack using time-series techniques. The culmination of this analysis is relayed to the controller, which employs event correlation techniques to discern and classify DDoS traffic from regular network activity.

Conclusion

This work introduces an RDAER model that integrates multiple techniques within the context of SDN to proactively identify and address DDoS attacks in different SDN-based cloud environments. This approach involves feature selection, clustering, time series analysis, and event correlation-based classification to enhance early detection of DDoS anomalies in network traffic. By examining each OpenFlow switch individually, RDAER's five-module structure enables effective data preprocessing, and selects key features USIA and NUDIA using RFE. These selected features are then grouped into clusters, considering their timestamps, to facilitate

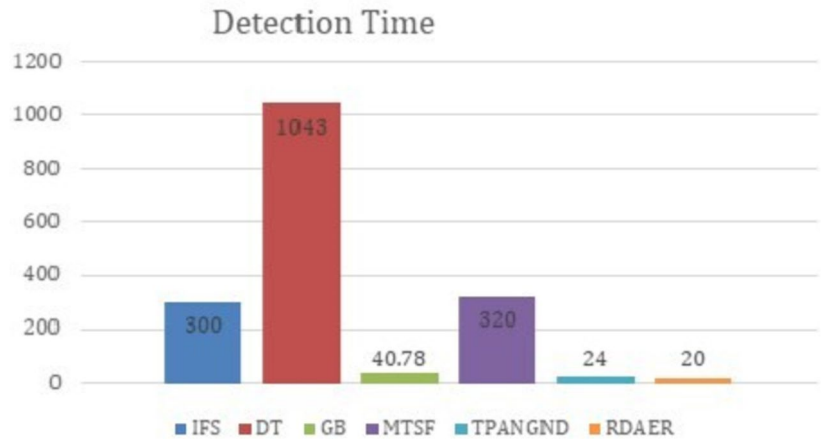

Fig. 17 Detection time comparison

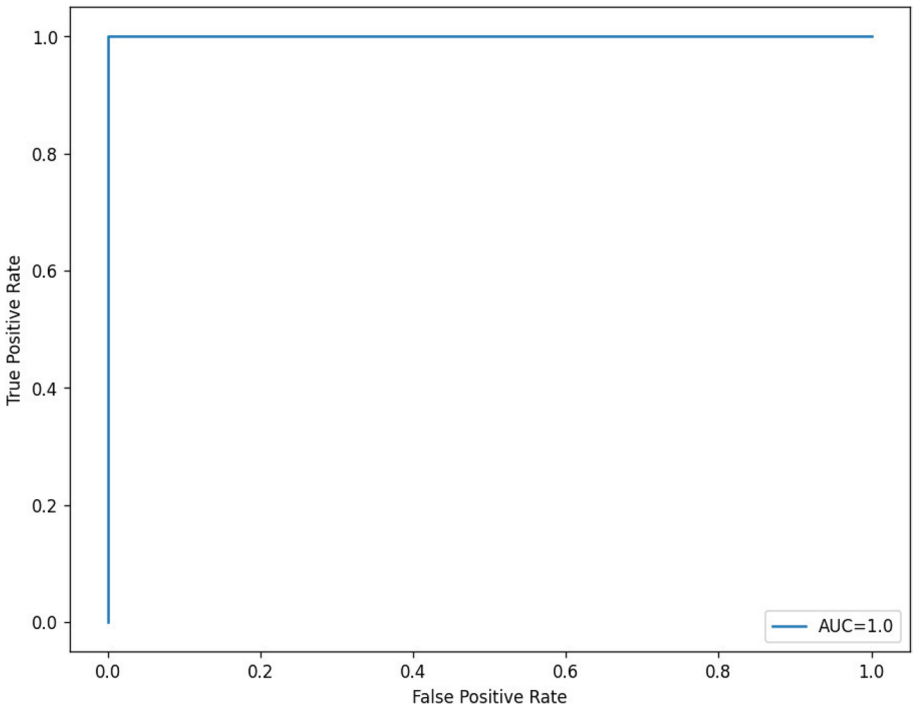

Fig. 18 The ROC curve

comprehensive traffic analysis. Within each cluster, a range of techniques, including ARIMA, Lyapunov exponent, exponential smoothing, and dynamic threshold calculations, is applied to compute two scores: score1 and score2. At the controller level final scores are calculated and correlated using rule-based classifier to classify traffic as either DDoS or normal. A DDoS attack warning is also issued whenever a switch detects any instances of a DDoS attack, and the countermeasure module alters the flow table to block the attack. The proposed RDAER model achieved a high accuracy rate of 99.92% and a fast detection time of 20 s in detecting DDoS attacks. In the future, the RDAER training and testing data can be further tuned to improve accuracy.

Authors' contributions
Asha Varma Songa: Conceived the concept that was put forward, studied the concepts used in this work, developed the algorithm, performed the experiments, discussed results, incorporated comments, and improved the research. He/ she also wrote the manuscript with the help of the guide.
Ganesh Reddy Karri: Conceived the idea, verified the methods, designed algorithms, gave feedback and improved the research, and helped in writing the manuscript.

Funding
The author(s) received no financial support for the research, authorship, and/ or publication of this article.

Availability of data and materials
The data can be provided if necessary.

Declarations

Ethics approval and consent to participate
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 28 November 2023 Accepted: 1 March 2024
Published online: 20 March 2024

References

1. Butt UA, Amin R, Mehmood M, Aldabbas H, Alharbi MT, Albaqami N (2023) Cloud security threats and solutions: a survey. Wireless Pers Commun 128(1):387–413
2. Sharma VK, Singh A, Jaya KR, Bairwa AK, Srivastava DK (2022) Introduction to virtualization in cloud computing," In Machine Learning and Optimization Models for Optimization in Cloud. Chapman and Hall/CRC. (pp. 1–14)
3. Alashhab ZR, Anbar M, Singh MM, Hasbullah IH, Jain P, Al-Amiedy TA (2022) Distributed denial of service attacks against cloud computing environment: survey, issues, challenges and coherent taxonomy. Appl Sci 12(23):12441
4. Ramprasath J, Seethalakshmi V (2021) Improved network monitoring using software-defined networking for ddos detection and mitigation evaluation. Wireless Pers Commun 116(3):2743–2757
5. Khorsandroo S, Sanchez AG, Tosun AS, Arco JM, Doriguzzi-Corin R (2021) Hybrid sdn evolution: a comprehensive survey of the state-of-the-art. Comput Netw 192:107981
6. Gadallah WG, Omar NM, Ibrahim HM (2021) Machine learning-based distributed denial of service attacks detection technique using new features in software-defined networks. Int J Comput Netw Inform Secur 13(3):15–27
7. Rawat SG, Obaidat MS, Pundir S, Wazid M, Das AK, Singh DP, Hsiao KF (2023) A Survey of DDoS Attacks Detection Schemes in SDN Environment. In 2023 International Conference on Computer, Information and Telecommunication Systems (CITS) (pp. 01–06). IEEE
8. Valdovinos IA, Perez-Diaz JA, Choo KKR, Botero JF (2021) Emerging ddos attack detection and mitigation strategies in software-defined networks: Taxonomy, challenges and future directions. J Netw ComputAppl 187:103093
9. Pandey P (2021) Security attacks in cloud computing
10. C¸agatay Ates¸, S¨ uleyman¨ Ozdel, and Emin Anarım, " Graph–based anomaly detection using fuzzy clustering," In *International Conference on Intelligent and Fuzzy Systems*, pp. 338–345, 2019
11. Raj MG, Pani SK (2021) A meta-analytic review of intelligent intrusion detection techniques in cloud computing environment. Int J Adv Comput Sci Appl 12(10):206–217
12. Dong S, Abbas K, Jain R (2019) A survey on distributed denial of service (DDoS) attacks in SDN and cloud computing environments. IEEE Access 7:80813–80828
13. Dong S, Sarem M (2019) DDoS attack detection method based on improved KNN with the degree of DDoS attack in software-defined networks. IEEE Access 8:5039–5048
14. Dahiya A, Gupta BB (2020) Multi attribute auction based incentivized solution against ddos attacks. Comput Secur 92:101763
15. MahdaviHezavehi S, Rahmani R (2020) An anomalybased framework for mitigating effects of ddos attacks using a third party auditor in cloud computing environments. Cluster Comput 23(4):2609–2627
16. Sadeghpour S, Vlajic N, Madani P, Stevanovic D (2021) Unsupervised ML based detection of malicious web sessions with automated feature selection: Design and real-world validation. In 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), pp. 1–9, IEEE
17. Lee S, Kim G, Kim S (2011) Sequence-order-independent network profiling for detecting application layer ddos attacks. EURASIP J Wirel Commun Netw 2011:1–9
18. Ribeiro MA, Fonseca MSP, de Santi J (2023) Detecting and mitigating DDoS attacks with moving target defense approach based on automated flow classification in SDN networks. Comput Secur 134:103462
19. Venkatesh B, Anuradha J (2019) A review of feature selection and its methods. Cybernetics Inform Technol 19(1):3–26
20. Fouladi RF, Ermis O, Anarim E (2020) A ddos attack detection and defense scheme using time-series analysis for sdn. J Inform Secur Appl 54:102587
21. Karthick MK, Kiruthiga G, Saraswathi PM, Dhiyanesh B, Radha R (2022) A subset scaling recursive feature collection based DDoS detection using behavioural based ideal neural network for security in a cloud environment. Procedia Computer Science 215:509–518
22. Alubaidan H, Alzaher R, AlQhatani M, Mohammed R (2023) Ddos detection in Software-Defined Network (Sdn) using machine learning. Int J Cybernetics Inform 12:4
23. Samaan SS, Jeiad HA (2023) Feature-based real-time distributed denial of service detection in SDN using machine learning and Spark. Bullet Electric Eng Inform 12(4):2302–2312
24. Zhou H, Zheng Y, Jia X, Shu J (2023) Collaborative prediction and detection of DDoS attacks in edge computing: a deep learning-based approach with distributed SDN. Comput Netw 225:109642
25. Najafimehr M, Zarifzadeh S, Mostafavi S (2022) A hybrid machine learning approach for detecting unprecedented ddos attacks. J Supercomputing 1–31
26. Dinh PT, Park M (2020) ECSD: Enhanced compromised switch detection in an SDN-based cloud through multivariate time-series analysis. IEEE Access 8:119346–119360
27. Peng H, Sun Z, Zhao X, Tan S, Sun Z (2018) A detection method for anomaly flow in software defined network. IEEE Access 6:27809–27817
28. Jose Su´ arez-Varela and Pere Barlet-Ros (2018) Flow monitoring in software-' defined networks: finding the accuracy/performance tradeoffs. Comput Netw 135:289–301
29. Manjunath CR, Rathor K, Kulkarni N, Patil PP, Patil MS, Singh J (2022) Cloud based DDOS attack detection using machine learning architectures: understanding the potential for scientific applications. Int J Intell Syst Appl Eng 10(2s):268–271
30. Aamir M, Ali Zaidi SM (2021) Clustering based semisupervised machine learning for ddos attack classification. J King Saud Univ Comput Inform Sci 33:436–446
31. Hajimaghsoodi M, Jalili R (2022) Rad: a statistical mechanism based on behavioral analysis for ddos attack countermeasure. IEEE Trans Inf Forensics Secur 17:2732–2745
32. Raja Sree T, Mary SairaBhanu S (2020) Detection of http flooding attacks in cloud using fuzzy bat clustering. Neural Comput Appl 32:9603–9619
33. Girish L, Rao SK (2021) Anomaly detection in cloud environment using artificial intelligence techniques. Computing 1–14
34. Zelaya C. V. G. (2019). Towards explaining the effects of data preprocessing on machine learning. In 2019 IEEE 35th international conference on data engineering (ICDE) (pp. 2086–2019). IEEE
35. Lian W, Nie G, Jia B, Shi D, Fan Qi, Liang Y (2020) An in trusion detection method based on decision tree-recursive feature elimination in ensemble learning. Math Probl Eng 2020:1–15
36. Lonnie Shumirai Matsa, Guy-Alain Zodi-Lusilao, and Fungai Bhunu Shava, "Recursive feature elimination for ddos detection on software define network," In 2021 IST-Africa Conference (IST-Africa), pp. 1– 10. IEEE, 2021

37. Kumar R, Kumar P, Kumar Y (2022) Multi-step time series analysis and forecasting strategy using arima and evolutionary algorithms. Int J Inf Technol 14(1):359–373
38. S de O Domingos, Joao FL de Oliveira, and Paulo SG de Mattos Neto (2019) An intelligent hybridization of arima with machine learning models for time series forecasting. Knowledge-Based Systems 175:72–86
39. Jain G, Mallick B (2017) A study of time series models arima and ets. Environ Anthropol eJournal
40. Jun MA (2022) Chaos theory and applications, the physical evidence, mechanism are important in chaotic systems. Chaos Theor Appl 4(1):1–3
41. Sharafaldin I, Lashkari AH, Hakak S, Ghorbani AA (2019) Developing realistic distributed denial of service (ddos) attack dataset and taxonomy," In 2019 International Carnahan Conference on Security Technology (ICCST), pp. 1–, IEEE
42. Ring M, Wunderlich S, Scheuring D, Landes D, Hotho A (2019) A survey of network-based intrusion detection data sets. Comput Secur 86:147–167
43. Yonghao Gu, Li K, Guo Z, Wang Y (2019) Semisupervised k-means ddos detection method using hybrid feature selection algorithm. IEEE Access 7:64351–64365
44. Jasim MN, Gaata MT (2022) K-Means clustering-based semi-supervised for DDoS attacks classification. Bullet Electric Eng Inform 11(6):3570–3576
45. Liu X, Ren J, He H, Wang Q, Song C (2021) Lowrate ddos attacks detection method using data compression and behavior divergence measurement. Comput Secur 100:102107
46. David J, Thomas C (2020) Detection of distributed denial of service and existing methods attacks based on information theoretic approach in time series models. J Inform Secur Appl. 55:102621
47. Maheshwari V, Bhatia A, Kumar K (2018) Faster detection and prediction of ddos attacks using mapreduce and time series analysis. In 2018 International Conference on Information Networking (ICOIN), pp 556–561
48. Alghawli AS (2022) Complex methods detect anomalies in real time based on time series analysis. Alex Eng J 61(1):549–561
49. Kushwah GS, Ranga V (2020) Voting extreme learning machine based distributed denial of service attack detection in cloud computing. J Inform Secur Appl 53:102532
50. Shyla SI, Sujatha SS (2020) Cloud security: LKM and optimal fuzzy system for intrusion detection in cloud environment. J Intell Syst 29(1):1626–1642
51. SaiSindhuTheja R, Shyam GK (2021) An efficient metaheuristic algorithm based feature selection and recurrent neural network for dos attack detection in cloud computing environment. Appl Soft Comput 100:106997
52. Prathyusha DJ, Kannayaram G (2021) A cognitive mechanism for mitigating ddos attacks using the artificial immune system in a cloud environment. Evol Intell 14(2):607–618
53. Bhardwaj A, Mangat V, Vig R (2020) Hyperband tuned deep neural network with well posed stacked sparse autoencoder for detection of ddos attacks in cloud. IEEE Access 8:181916–181929
54. Hussain YS (2020) Network intrusion detection for distributed denial ofservice (ddos) attacks using machine learning classification techniques
55. D.V.V.S. Manikumar and B Uma Maheswari. Blockchain based ddos mitigation using machine learning tech- niques. In 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), pages 794–800, 2020. doi: https://doi.org/10.1109/ICIRCA48905.2020.9183092
56. Rajagopal S, Kundapur PP, Hareesha KS (2021) Towards effective network intrusion detection: From concept to creation on azure cloud. IEEE Access 9:19723–19742. https://doi.org/10.1109/ACCESS.2021.3054688
57. Khoei TT, Aissou G, Hu WC, Kaabouch N (2021) Ensemble learning methods for anomaly intrusion detection system in smart grid. In 2021 IEEE International Conference on Electro Infor- mation Technology (EIT). 129–135. https://doi.org/10.1109/EIT51626.2021.9491891
58. Almiani M, Abughazleh A, Jararweh Y, Razaque A (2022) Resilient back propagation neural network security model for containerized cloud computing. Simul Model Pract Theory 118:102544
59. Marvi M, Arfeen A, Uddin R (2021) A generalized machine learning-based model for the detection of ddos attacks. Int J Netw Manage 31(6):e2152
60. Aytac T, Ali Aydın M, Zaim AH (2020) Detection of ddos attacks using machine learning methods
61. Batchu RK, Seetha H (2021) A generalized machine learning model for DDoS attacks detection using hybrid feature selection and hyperparameter tuning. Comput Netw 200:108498
62. Daffu P, Kaur A (2016) Mitigation of ddos attacks in cloud computing. In 2016 5th International Conference on Wireless Networks and Embedded Systems (WECON), pages 1–5. IEEE

Publisher's Note

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).
Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.
These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.
We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.
While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.
These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.
To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.
Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.
If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com