Measurement: Sensors 28 (2023) 100828

Contents lists available at ScienceDirect

Measurement: Sensors

journal homepage: www.sciencedirect.com/journal/measurement-sensors

LRDADF: An AI enabled framework for detecting low-rate DDoS attacks in cloud computing environments

M.Jahir Pasha [a], K.Prasada Rao [b], A. MallaReddy [c,*], Vasavi Bande [d]

[a] Department of Computer Science & Engineering, G. Pullaiah College of Engineering and Technology, Kurnool, Andhra Pradesh, India
[b] Department of CSE, GITAM School of Technology, GITAM University, Visakhapatnam, Andhra Pradesh, India
[c] Department of Information Technology, CVR College of Engineering, Hyderabad, Telangana, India
[d] Department of Information Technology, MVSR Engineering College, Nadergul, Hyderabad, Telangana, India

ARTICLE INFO

Keywords:
Low-rate DDoS attacks
Cloud computing
Attack mitigation
Container based cloud infrastructure

ABSTRACT

DDoS attacks, also known as distributed denial-of-service attacks, pose a significant risk to networks in the cloud. The attackers aim to flood the target system with an overwhelming amount of data and requests until it becomes completely overloaded and unable to function properly. These attacks are becoming smarter and more dangerous all the time. A low-rate DDoS attack is one such strategy that makes detection difficult. At the same time, cloud infrastructure is rapidly evolving. Container-based technology makes it possible for cloud computing to use resources efficiently and scale services in a flexible way. Existing methods for detecting DDoS attacks in cloud computing are insufficient when adversaries use low-rate DDoS attacks. A method is required that can not only identify the attack but also prevent it to some extent. A Low-Rate DDoS Attack Detection Framework (LRDADF) was proposed for this purpose when adversaries use low-rate DDoS attacks. A comprehensive approach is required because low-rate DDoS attacks are difficult to detect. In addition to employing deep learning methods to detect such attacks, we proposed a mathematical model to realize a mitigation strategy. As a result, we proposed a new algorithm called the Hybrid Approach for Low-Rate DDoS Detection (HA-LRDD). The algorithm employs an AI-enabled method comprised of deep convolutional neural networks (CNN) and a deep auto encoder. We defined another algorithm called Dynamic Low-Rate DDoS Mitigation (DLDM), which mitigates the impact of an attack once it has been identified. It also ensures that the attack is defeated and that the infrastructure continues to operate. A comprehensive simulation study revealed that the proposed framework is capable of detecting and mitigating low-rate DDoS attacks to ensure an acceptable level of service in cloud computing environments.

1. Introduction

In the past few years, there has been a surge in Distributed Denial of Service (DDoS) attacks on cloud-based services due to the widespread use of cloud computing [1]. These attacks are a form of cyber-attack where an attacker floods a server, website, or network with excessive traffic or requests to interrupt its regular functioning. As a result, legitimate users are unable to access the service, causing significant harm, particularly in cloud computing environments where multiple services rely on one another.

The detection of low-rate DDoS attacks in cloud computing environments is a challenging task due to the large-scale and dynamic nature of cloud environments [2], the wide range of possible attack vectors, and the need to balance detection accuracy with minimal impact on legitimate traffic. Traditional signature-based detection methods are not effective against these attacks as they rely on identifying known attack patterns, which are often modified by attackers to avoid detection. Moreover, cloud environments are highly heterogeneous, consisting of various devices with different capabilities and resources, making it difficult to deploy a uniform defense mechanism across the entire environment.

The main problem addressed in this research is the detection of low-rate DDoS attacks in cloud computing environments. Low-rate attacks are particularly challenging to detect as they involve a small amount of traffic that is spread over a long period, making it difficult to distinguish from legitimate traffic [3]. The proposed solution aims to detect these

* Corresponding author.
E-mail addresses: jahir4444@gmail.com (M.Jahir Pasha), pkaru@gitam.edu (K.Prasada Rao), mallareddyadudhodla@gmail.com (A. MallaReddy), vasavi.bande@gmail.com (V. Bande).

https://doi.org/10.1016/j.measen.2023.100828
Received 18 October 2022; Received in revised form 12 May 2023; Accepted 12 June 2023
Available online 21 June 2023

attacks by leveraging machine learning techniques and network traffic analysis to identify anomalous traffic patterns that are indicative of a DDoS attack.

The reason for conducting this research is to enhance the safety of cloud computing environments by implementing an effective and efficient system for identifying low-rate DDoS attacks. These attacks have the potential to cause substantial harm to cloud services, resulting in negative publicity and financial loss for both the cloud service providers and their customers. The solution proposed is to create an AI-based framework capable of detecting these attacks instantly, thereby minimizing their impact and bolstering the overall security of cloud environments.

In cloud computing environments, various devices are considered, including servers, virtual machines, storage devices, and networking devices [4]. These devices have different capabilities and resources, and they are often provided by different vendors, leading to a highly heterogeneous environment. The heterogeneity of cloud environments poses a challenge for developing a uniform defense mechanism that can be applied to all devices. The proposed solution aims to address this challenge by providing a device-agnostic framework that can be applied to various devices in the cloud environment.

The heterogeneity of cloud environments refers to the diversity of devices, platforms, and services that make up the environment [5]. This heterogeneity makes it challenging to develop a uniform defense mechanism that can be applied to all devices. The proposed solution aims to address this challenge by providing a device-agnostic framework that can be applied to various devices in the cloud environment. This framework leverages machine learning techniques to identify anomalous traffic patterns and can be applied to different devices, regardless of their capabilities or resources [6].

The device request specification refers to the process of specifying the requirements of a device that will be used to support the proposed solution. These requirements include the hardware and software specifications, network connectivity, and other resources needed to deploy and operate the solution. The device request specification process is critical to ensuring that the solution can be deployed effectively and efficiently across the cloud environment [7]. The proposed solution aims to provide a flexible and scalable approach to detecting low-rate DDoS attacks in cloud computing environments. Therefore, the device request specification needs to consider the heterogeneity of devices within the cloud environment [8]. This refers to the fact that the cloud infrastructure comprises a variety of devices with different hardware specifications, software versions, and network configurations. The solution needs to be able to operate on this heterogeneous infrastructure seamlessly. Hence, the device request specification process needs to take into account the different types of devices that will be considered for the solution, their capabilities, and limitations [9]. This will enable the proposed solution to be optimized for each device, leading to better performance and improved detection accuracy.

In this paper, we're suggesting a technique that relies on deep learning, specifically using CNN and deep autoencoders. Our contributions can be summarized as follows.

1. The paper proposes a framework for detecting low-rate Distributed Denial of Service (DDoS) attacks using a combination of Sparse Autoencoder (SAE) and Convolutional Neural Network (CNN).
2. The proposed framework is implemented using an algorithm called Hybrid Approach for Low-Rate DDoS Detection (HA-LRDD), which involves feature extraction, normalization, deep neural networks, and classification.
3. The paper introduces two evaluation metrics, detection rate and false positive rate, to compare the proposed method with existing methods and demonstrates the effectiveness of the proposed method.

This paper introduces a new framework and algorithm that can detect low-rate DDoS attacks. By using an autoencoder, dropout techniques, and CNN, the proposed method shows better performance than previous methods. The evaluation metrics used provide a comprehensive analysis of the proposed method's performance.

The paper is organized as follows: Section 2 covers literature on different types of attacks and solutions, providing valuable insights and identifying research gaps. Section 3 introduces the proposed system, while Section 4 presents experimental results and performance evaluation. Finally, Section 5 concludes the paper and provides directions for future research.

2. Related work

This section provides a literature review of the tools that can be used to detect DDoS attacks in the cloud. It also elucidates low-rate.

2.1. DDoS attack detection in cloud environments

In view of cloud computing, DDoS attackers shifted their targets to cloud environments. Sharma et al. [10] explored DoS attacks in the cloud and defined a method based on Shannon's entropy for detection. In addition to discarding the traffic, it could detect such attacks and generate an alarm. Srilakshmi p et al. [11] investigated the DDoS attack defence mechanisms for cloud environments. They discussed different kinds of DDoS attacks, such as reflective attacks, spoofing, web-service addressing, coercive parsing, port scanning, user-to-root attacks, spoofing, and flooding. They classified attacks based on the IDS approach, scalability, user authentication, and response mechanism. They intend to conduct additional research on VM attacks in the cloud. Agrawal and Tapaswi [12] studied methods to defend DDoS attacks in the cloud and observed certain challenges in defence mechanisms. The open challenges they found include the heterogeneity of botnet devices in IoT use cases, the vulnerability of SDNs, cheaper means of creating botnets, and the presence of DeNy attacks in the cloud.

Bhushan and Gupta [13] explored SDN-based solutions to detect DDoS attacks in cloud environments. They exploited the features of SDN in order to have a better and more focused approach towards attack detection. Their architecture has different components for data, application, and control planes. Hezavehi and Rahmani [14] proposed a third-party auditing approach in their anomaly-based DDoS attack detection framework. Anomaly auditing was carried out by third-party auditors. It makes use of deviations in response times in order to detect attack scenarios. Wani et al. [15] investigated ML approaches for the detection of DDoS attacks. Their attack model has a botnet with zombies in order to analyse the proposed method. Hu et al. [16] defined a system known as DDoS Flooding Attack Detection and Mitigation (FADM). It is in the presence of an SDN-based controller that collects traffic patterns. An entropy-based measure and ML techniques like SVM are used to detect attacks. In the future, they intend to improve it to detect DDoS attacks at the application layer.

Buragohain and Medhi [17] proposed an SDN-based system to detect such attacks in the cloud. They employed a 4-port tree topology in the empirical study. Their solution is named FlowTrApp. Bhushan and Gupta [18] used an SDN-assisted cloud to propose a DDoS attack detection method. In such clouds, they observed many security vulnerabilities linked to scalability, availability, access control, threats from applications, fraudulent flow rules, authentication and authorization, DDoS attacks, and flow table overloading. Their approach could protect SDN from DDoS attacks.

Kushwah and Ranga [19] provided a system to detect DDoS attacks. It is an approach based on voting machines with extreme learning capabilities. It is a kind of artificial neural network (ANN) used to detect network attack traffic effectively. Sahi et al. [20] developed a system to detect flooding attacks that deplete cloud resources. Their solution has detection and prevention mechanisms. It could reduce the chances of attacks and optimize resource utilization. Dong et al. [21] explored SDN architecture for the cloud and its vulnerabilities, along with defence

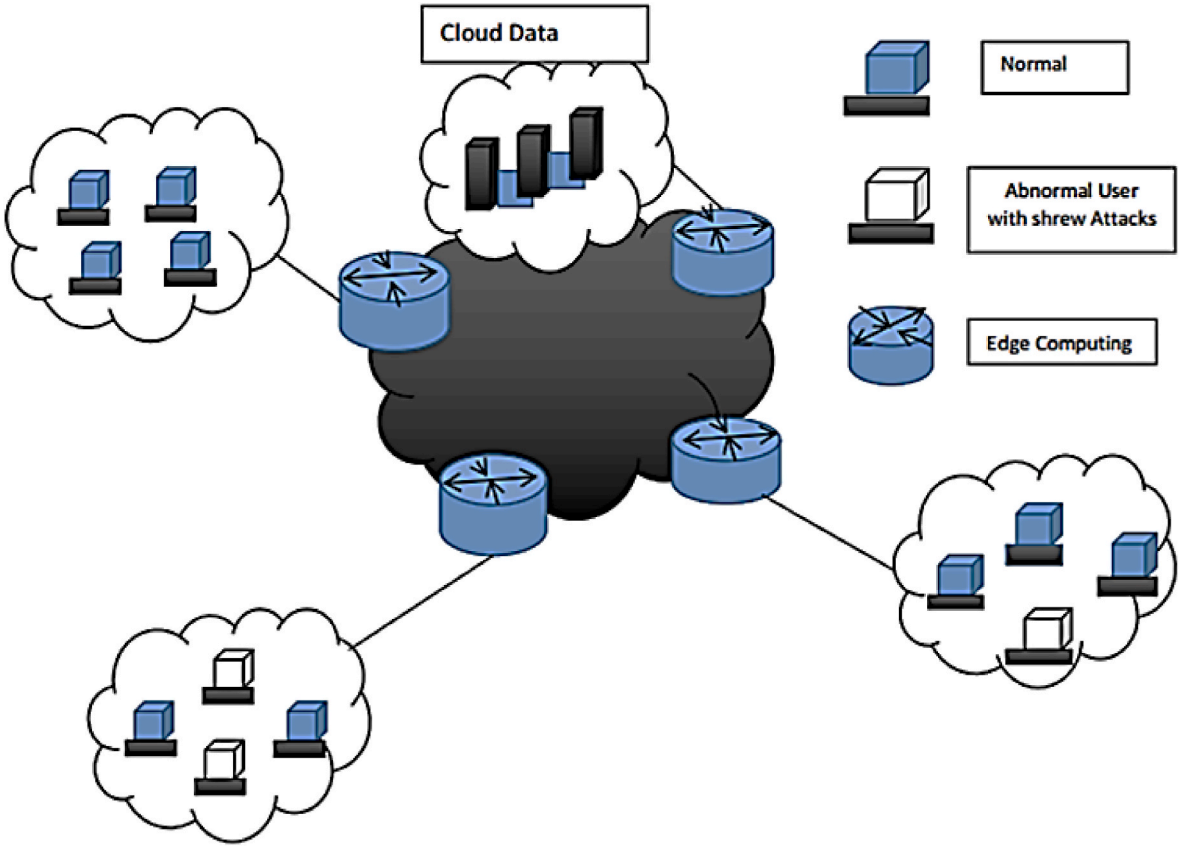

Fig. 1. The system model showing the problem context for low rate DDoS attacks.

mechanisms against DDoS attacks. Tsai et al. [22] do similar research that focuses more on VM attacks. Somani et al. [23] investigated methods to combat DDoS attacks. The methods include attack filtering, mitigation through time, and collaboration approaches. They anticipated trouble with DDoS attacks in the future due to the emerging "DDoS for hire" services.

Liu et al. [24] proposed an algorithm known as BIRTH based on a novel network flow grouping method. It could prevent periodic swarm DDoS attacks. Autocorrelation of frequency domain characteristics is used to achieve this. Kamboj et al. [25] studied various DDoS attacks, including distributed reflective denial of service (DRDoS). Prathyusha and Kannayaram [26] proposed an artificial immune system that is based on a cognitive procedure to mitigate DDoS attacks. Rios et al. [27] combined ML and fuzzy logic to propose a method to detect reduction-of-quality DDoS attacks. Osanaiye et al. [28] explored DDoS attacks in the cloud and mitigation strategies.

2.2. Dealing with low rate DDoS attacks

DDoS attacks exhibit large volumes of traffic. Low-rate DDoS attacks, on the other hand, maintain low traffic, making detection difficult. Perez-Diaz et al. [29] explored a software-defined network (SDN)-based architecture for detecting and mitigating DDoS attacks. Their solution is based on machine learning techniques. They trained an IDS in order to have protection from DDoS attacks. Their method could produce higher accuracy in detecting low-rate DDoS attacks and mitigating them. Agrawal and Tapaswi [30] proposed a lightweight method for detecting low-rate DDoS attacks belonging to the IP spoofing category. It is an adaptive approach that could handle high-rate DDoS attacks as well in an Eucalyptus cloud environment. In the future, they intend to focus on differentiating legitimate packets from spoof packets.

Bhushan and Gupta [31] proposed a hypothesis and tested it with respect to low-rate DDoS attacks in cloud environments. Attack representation and detection processes are discussed in detail. It could differentiate between an attack period and a non-attack period. Their algorithm cloud detects attacks with a traffic sampling rate and threshold. Liu et al. [32] proposed a methodology using edge computing and deep CNN-based Q-Learning to detect low-rate DDoS attacks. In the process, they also used ML techniques like SVM to classify traffic patterns. They intended to improve it further to deal with sparse traffic as well. Zhang et al. [33] proposed a PSD and ML-based technique. PSD entropy and adaptive threshold are used to achieve this effectively. Sahoo et al. [34] used information-distance metrics to detect attacks in SDN-based cloud data centers. An OpenFlow SDN controller is used for empirical study. In an SDN-based cloud computing scenario, they could look into security issues at the control plane.

Yevsieieva and Helalat [35] focused on finding the effects of low-rate DDoS attacks on cloud environments. Zhou et al. [36] proposed a method based on the expectation of packet size to detect low-rate DDoS attacks. They used it as a measure to distinguish between good traffic and malicious traffic pertaining to pulsing attacks and constant attacks. Different tolerance factors are used in order to have an effective solution. Liu et al. [37] proposed a method based on a measure known as behavior divergence and data compression to ascertain the presence of low-rate DDoS. Agrawal and Tapaswi [38] defined a defence against such attacks based on "power spectral density analysis." By monitoring traffic patterns, their method could identify the attacks. Wu et al. [39] proposed an algorithm based on sequence alignment detection for detecting synchronous low-rate DDoS attacks. Literature has shown that there have been methods to detect low-rate DDoS attacks. However, supervised machine learning is not suitable for such attacks due to inadequate training samples. In this paper, we propose a method based on deep learning, particularly CNN and deep autoencoders.

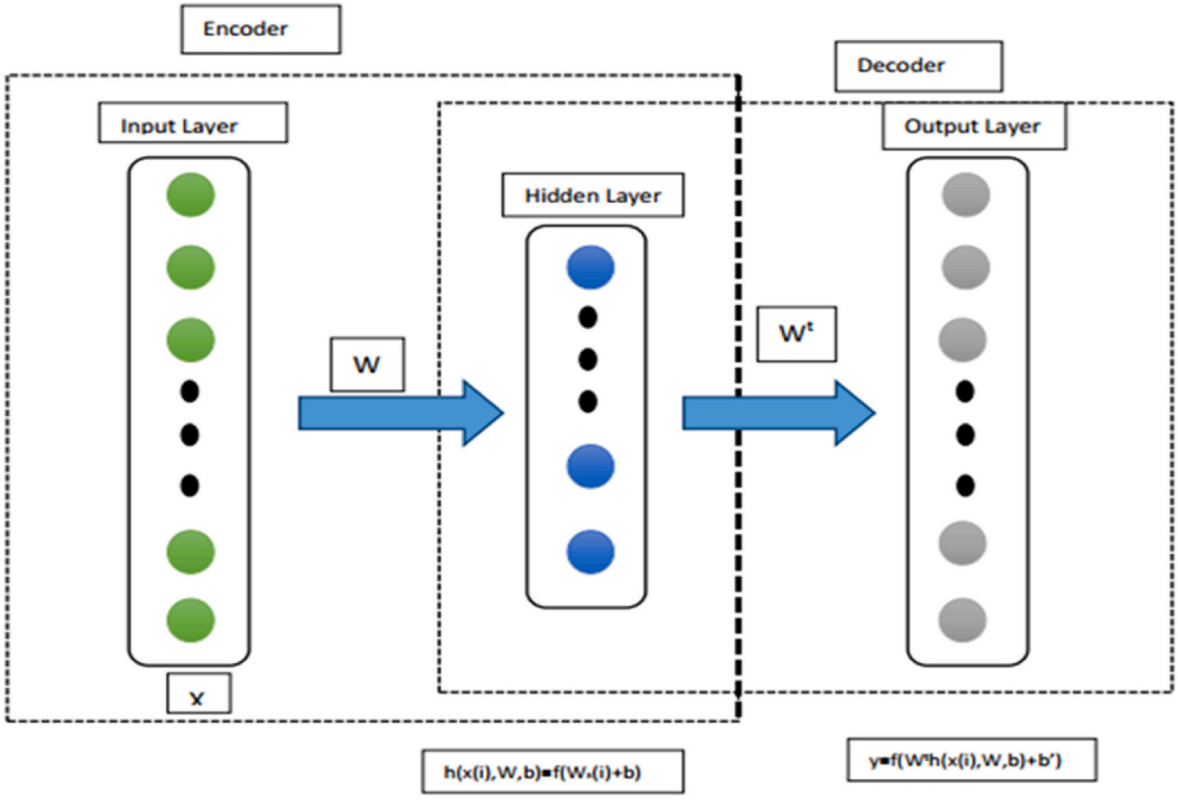

Fig. 2. Deep autoencoder structure.

3. Proposed methodology

The proposed methodology is described in the following subsections that focus on the attack model, the proposed framework, and algorithm and evaluation procedure.

3.1. The attack model

With low-rate DDoS attacks, the attacker performs attacks at a low cost. However, it comes at the cost of wasting bots' capabilities. In the presence of edge computing, there are more chances of being vulnerable to low-rate DDoS attacks. Fig. 1 shows a system model showing the context of the problem for low-rate DDoS attacks (see Fig. 2).

As presented in Fig. 1, the system model includes traditional cloud infrastructure and also edge computing resources. There are normal users and also abnormal users who launch shrewd attacks.

3.2. Deep autoencoder

The deep autoencoder has mechanisms for encoding and decoding. The former generates low-dimensional space from its high-dimensional space for given data. This transformation of data is useful to arrive at features that contribute to the detection of attack traffic. In the presence of autoencoders, regularization and dimensionality reduction have a role to play. Particularly, regularization identifies nodes that have a high impact on the outcomes of the network. An approximation process is learned at the hidden layer while the data reconstruction takes place at the output layer.

A simple autoencoder is aimed at reducing loss, improving model accuracy, and speeding up convergence. The autoencoder learns from traffic patterns and helps in recognizing attack traffic. The extracted statistical features help in the detection of such traffic. A min-max-based approach shown in Eq. (1) is used for normalization.

$$x_{norm} = \frac{x_i - x_{min}}{x_{max} - x_{mi\ n}} \tag{1}$$

Where x_{norm} is the normalized value x_i is a feature's value; x_{min} and x_{max} are minimum and maximum values respectively.

3.3. Proposed framework

The framework for detecting low-rate DDoS attacks using a combination of sparse auto-encoders (SAE) and convolutional neural networks (CNN) consists of the following steps.

1. **Data collection:** The initial step is to gather information from the network traffic, comprising both regular and offensive traffic. The information is accumulated in the time zone, which indicates that it is documented over a particular period of time.

Dataset used: The CIC-DDoS2019 dataset has been used to perform step 1 of the DDoS detection process. The CIC-DDoS2019 dataset is a publicly available dataset that contains network traffic traces generated in a controlled environment to simulate various DDoS attack scenarios. The dataset contains 15 different attack scenarios, with different types of attacks, traffic volume, and number of attacking machines. The first stage of identifying any potential DDoS activity includes utilizing datasets to teach a machine learning algorithm how to differentiate between regular and malware network traffic, during the process of model training on a particular dataset consisting of labeled data benign network activities are marked 'normal' whilst attacks such as DDoS are marked 'malicious'. The detection of DDoS attacks in real-time through classification of incoming traffic into benign or malignant categories is facilitated by the use of labeled data that helps learn normal and malicious network behaviors.

2. **Pre-processing:** so this involves removing noise and unwanted data from the initial dataset. It encompasses duties like discarding replicas of packets and sifting through minor traffic to normalize the data.
3. **Feature extraction:** to extract appropriate features. This case involves the extraction of time-domain characteristics like mean value along with statistical properties including variance and standard deviation, so to calculate these features we can use mathematical equations.
   - **Mean:** $(\frac{1}{N}) * sum(x)$, where N is the number of data points and x is the data value
   - **Variance:** $(\frac{1}{N}) * sum((x - mean)^2)$, where N is the number of data points, x is the data value, and mean is the mean value
   - **Standard deviation:** $sqrt(variance)$.
   - **Maximum value:** $\max(x)$.
   - **Minimum value:** $\min(x)$.
   - **Signal entropy:** $sum(p * \log(p))$, where p is the probability of each data value
4. **SAE feature learning:** Passing through a sparse autoencoder (SAE), the extracted features are learned and the compression and reconstruction of input data is realized through minimizing reconstruction errors in an SAE neural network. To achieve this result, the input data is transformed into a lower-dimensional space and then converted back to its original form, so during the encoding process of data by SAE's, it learns to identify important features that are necessary for detecting DDoS attacks.

Mathematical model of the SAE feature learning:

Let X be the input data with dimensionality d_x and N samples, represented as $X \in \mathbb{R}^{d_x x\ N}$.

The SAE consists of an encoder function f and a decoder function g, with a bottleneck layer in between, represented as h.

The encoder function takes the input data X and maps it to a lower-dimensional representation, $h = f(X)$, where $h \in \mathbb{R}^{d_h x\ N}$, with $d_h < d_x$.

The decoder function takes the lower-dimensional representation h and maps it back to the original space, $\widehat{X} = g(h)$, where $\widehat{X} \in \mathbb{R}^{d_x x\ N}$.

The goal of SAE feature learning is to learn the encoder and decoder functions such that the reconstruction error between the input data X and the reconstructed data $\widehat{X}$ is minimized.

This can be formulated as an optimization problem, where we want to find the parameters θ of the encoder and decoder functions that minimize the mean squared error (MSE) between X and $\widehat{X}$:

$$\min_{\theta} ||X - \widehat{X}||^2 \tag{2}$$

As the training process goes on, the encoder function f becomes adept at recognizing significant characteristics in the data that are necessary for detecting DDoS attacks. This is accomplished by compressing the input data into a space with fewer dimensions, while preserving the essential information.

Overall, the SAE feature learning process helps to extract useful features from the input data, which can then be used for DDoS attack detection.

5. **CNN classification:** The process of classifying involves passing the features learned by the SAE through a convolutional neural network (CNN) and the CNN is highly effective in identifying local patterns and features in data which makes it a powerful tool for image and signal processing tasks. CNN distinguishes between normal and attack traffic based on the features learned from input data through SAE.

Mathematical model:

Let X be the input data, which is a matrix of size N x M, where N is the number of samples and M is the number of features.

Let Y be the output, which is a vector of size N x 1, where each element represents the class label (normal or attack) for the corresponding input sample.

The CNN (Convolutional Neural Network) is made up of different layers such as convolutional layers, pooling layers, and fully connected layers. Let $C1, C2, ..., Cn$ be the convolutional layers, $P1, P2, ..., Pm$ be the pooling layers, and $F1, F2, ..., Fk$ be the fully connected layers.

The output of each layer is obtained by applying a set of filters or weights to the input. Let $W1, W2, ..., Wn$ be the filters for the convolutional layers, and Wk be the weights for the fully connected layers.

The output of the CNN can be represented as:

$$Y = softmax(Fk(hk)) \tag{3}$$

where hk is the output of the last fully connected layer, which is obtained by applying the weights Wk to the output of the previous layer. The softmax function is used to convert the output into a probability distribution over the two classes (normal and attack).

The input to the CNN is the features learned by the SAE. Let Z be the output of the SAE, which is a matrix of size N x L, where L is the number of hidden units in the SAE.

The output of each convolutional layer is obtained by applying a set of filters to the input. Let fi be the filter for the ith convolutional layer. Then the output of the ith convolutional layer is given by:

$$Ci = relu(conv(Z, fi)) \tag{4}$$

where $conv(Z, fi)$ is the convolution operation between Z and fi, and $relu(x)$ is the rectified linear unit activation function.

The output of each pooling layer is obtained by downsampling the input. Let pi be the pooling operation for the ith pooling layer. Then the output of the ith pooling layer is given by:

$$Pi = pi(Ci) \tag{5}$$

The output from the final pooling layer is flattened into a vector with one dimension (1D), and then fed into the fully connected layers to get the ultimate output. The CNN weights are taught using backpropagation and gradient descent, with the cross-entropy loss function used to calculate the loss between the predicted probabilities and the actual labels.

6. **Evaluation:** To assess the performance of the system several metrics such as accuracy and F1 score are employed which include precision and recall. Moreover, quantitative measurement of the system's ability to detect DDoS attacks can be done through time-domain analysis.

The proposed framework includes specific steps such as feature extraction, normalization, deep neural networks, and classification. First, the autoencoder learns from the data to understand its characteristics using unsupervised learning. As there is a feature engineering technique in the autoencoder, it can learn iteratively from the hidden features of given data. In the process, the autoencoder is capable of learning associations among features and gaining optimal knowledge in order to minimize features and extract representative features. Once features are learned, the autoencoder is trained. Random initialization of weight is made, and the setup of different variables such as sparse rate, dropout rate, denoising parameters, and learning rate is made. An average sparsity is computed. The sparse cost function is measured using Eq. (6), while and are updated using Eqs. (7) and (8), respectively.

$$C_{sparse}(w, b) = \frac{1}{n}\sum_{i=1}^{n} \|x_i - y_i\|^2 + \frac{\lambda}{2}\left(\sum_{l=1}^{l_s-1}\sum_{i-1}^{D}\sum_{j-1}^{C} W_{ij}(l) + \beta\sum_{j=1}^{C}\rho\ log\frac{\rho}{\rho_j} + (1-\rho)log\frac{1-\rho}{1-\rho_j}\right) \tag{6}$$

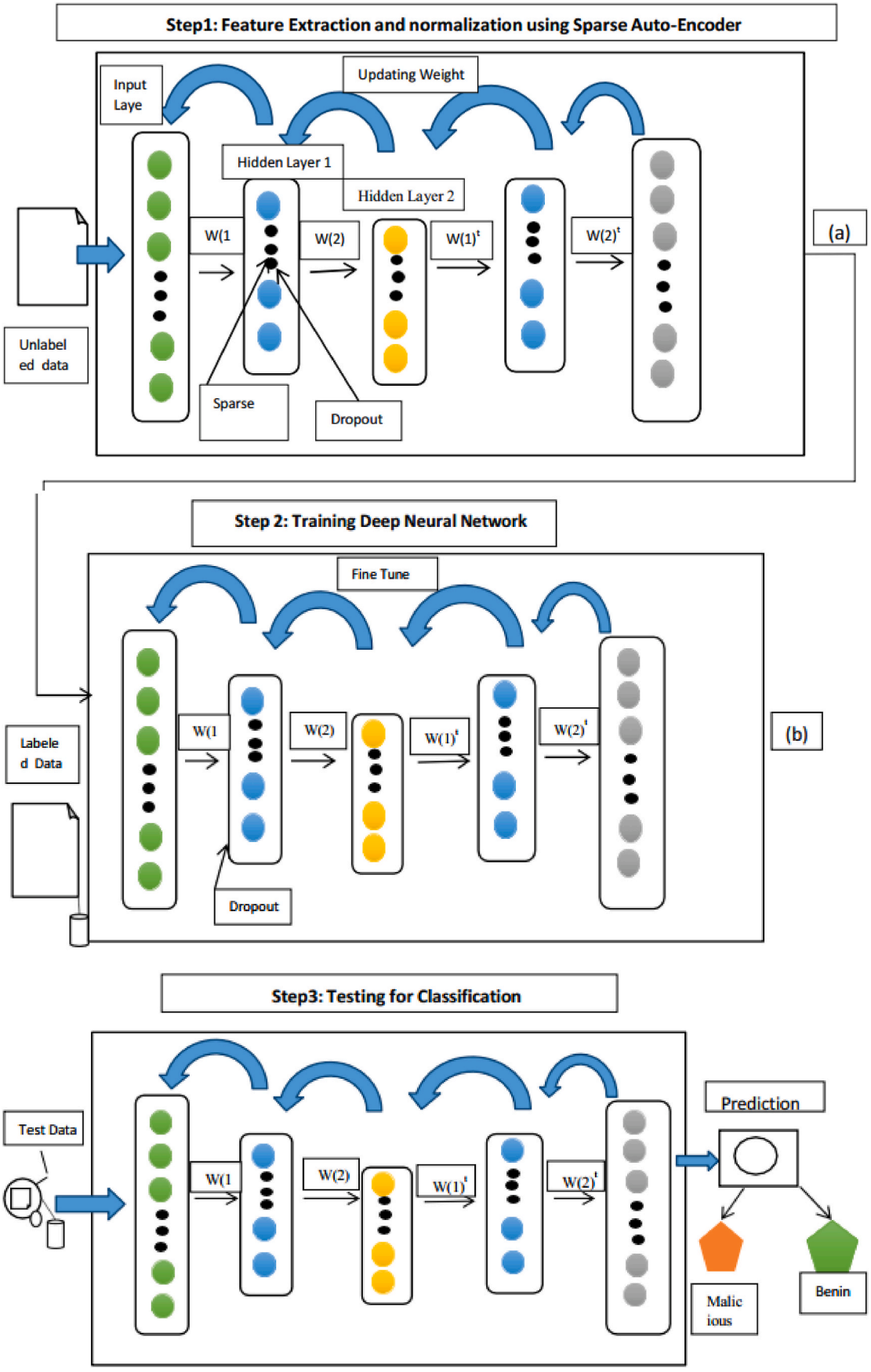

Fig. 3. Illustrates the complete framework based on simple autoencoder and CNN.

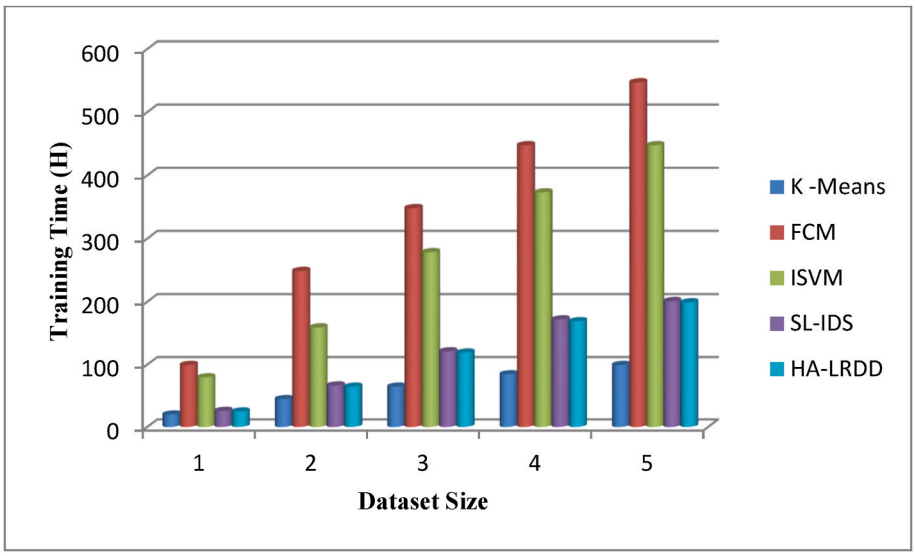

Fig. 4. Training time for different detection algorithms.

Table 1
List of variables and its meaning.

Variables	Meaning
x_norm	normalized value
x_i	feature's value
x_min	Minimum value
x_max	Max value
W,*b*	Random initialization parameters: W: weights between the input layer and the hidden layer; b : biases
n	number of data samples
D	Given dataset
a	activation of the hidden layer
DR	Data rate
FPR	False Positive rate
SGD	stochastic gradient descent
ρ_j	the average activation of each hidden neuron
SAE	Sparse Auto-Encoder
sr	sparse rate,
lr	Learning rate
dr	Dropout rate
T	test data
R	Low-rate DDoS attack detection results

Table 2
Simulation Parameters setting up of SAE-based DNN Methods.

SAE		
Parameter	Meaning	Value
M	Input nodes	2000
S_2	hidden nodes	600
out	output nodes	2000
ρ	sparse target	0.08
sr	sparse rate	0.4
lr	Learning rate	1
dr	Dropout rate	0.3
DNN		
M	Input nodes	2000
S	hidden nodes	600
Out	output nodes	2000
Dr	Dropout rate	0.3
Lr	Learning rate	1

Table 3
Shows training time of different algorithms against varied training data size.

Training Dataset Size	Training Time (m) of Different Algorithms				
	K -Means	FCM	ISVM	SL-IDS	HA-LRDD
1	19.9	99.5	79.6	25.87	24.875
2	44.775	248.75	159.2	66.665	64.675
3	64.675	348.25	278.6	121.39	119.4
4	84.575	447.75	373.125	172.135	169.15
5	99.5	547.25	447.75	200.99	199

Table 4
Shows performance of different algorithms in terms of detection rate and false positivity rate.

Detection Algorithms	Performance (%)	
	Detection Rate	False Positivity Rate
K-Means	74.99492	6.84315
FCM	78.32825	9.77022
ISVM	93.58349	0.47952
Sl-IDS	92.73264	1.34865
HA-LRDD	95.31522	0.56943

$$W_{ij} := W_{ij} - \in \frac{a}{a} C_{sparse}(W,b) \tag{7}$$

$$b_i := b_i - \in \frac{a}{a} C_{sparse}(W,b) \tag{8}$$

In Eq. (6), a sparse penalty is added to the underlying cost function. The parameters such as and must be defined while performing the coding process in the auto encoding. These two parameters are trained using the stochastic gradient descent (SGD) method. Eqs. (7) and (8) represent the method and the learning process, respectively. In order to minimize overfitting of the network, dropout is configured. The autoencoder-based CNN is subjected to dropout to leverage the learning process. CNN is initialized with parameters for a simple autoencoder. The network is fine-tuned adaptively from time to time. Finally, the framework CNN employs categorizes network traffic as benign or malicious. The complete framework is shown in Fig. 3.

As presented in Fig. 3, the framework has its functionality divided into three distinct steps (see Fig. 4). In step 1, a sparse autoencoder is used for normalization and feature extraction. The outcome of step 1 was used to train a deep neural network (DNN). Then, the outcome of

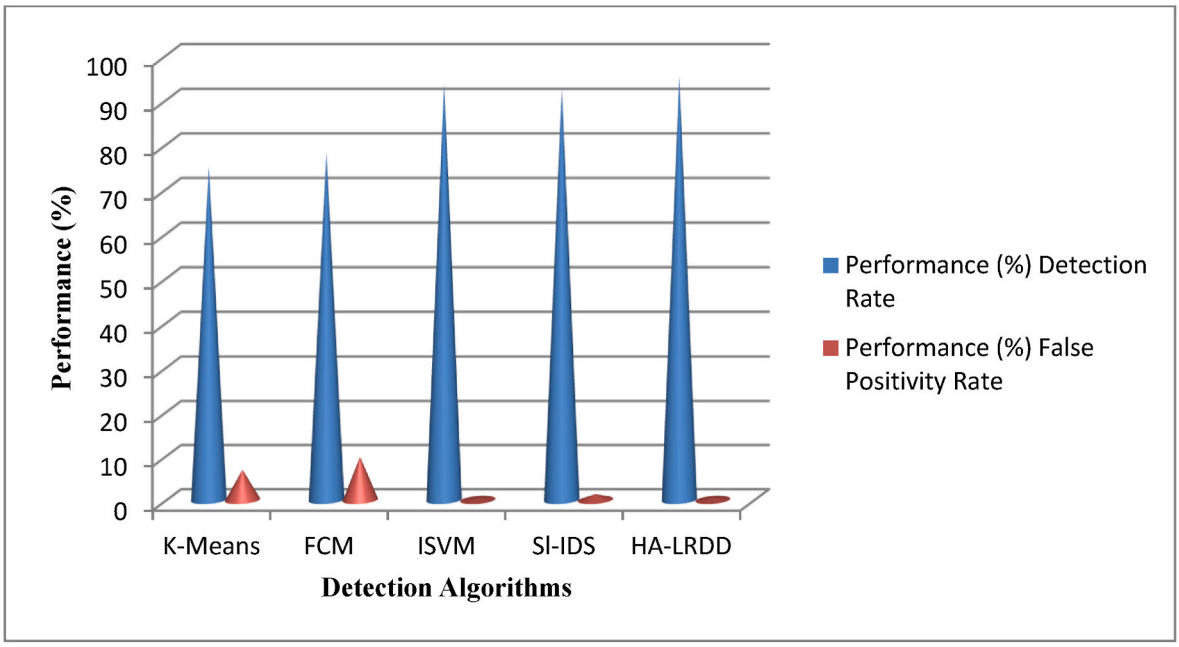

Fig. 5. Detection rate and false positivity rate comparison of different detection algorithms.

Table 5
Shows the congestion window in terms of attacks and deep learning model functioning to prevent attack.

Time (Sec)	Congestion Window	
	Attacks	Deep Learning
0	0	40
5	7	40
10	10	40
15	5	40
20	3	40
25	12	40
30	9	40

the second step is used in step 3 for the actual classification or prediction of class labels such as benign and malicious. Fig. 3(a) shows how an autoencoder is used with unsupervised learning to learn from data and provide features to DNN. The autoencoder has an iterative process to learn features from the hidden data. In other words, it is capable of feature engineering. It can extract more representative features by

Table 6
Shows the packet loss rate (%) in different network conditions.

Network Condition	Packet Loss Rate (%)
Normal	0.024
Attacks	0.06
Attacks while HA-LRDD Protection	0.032

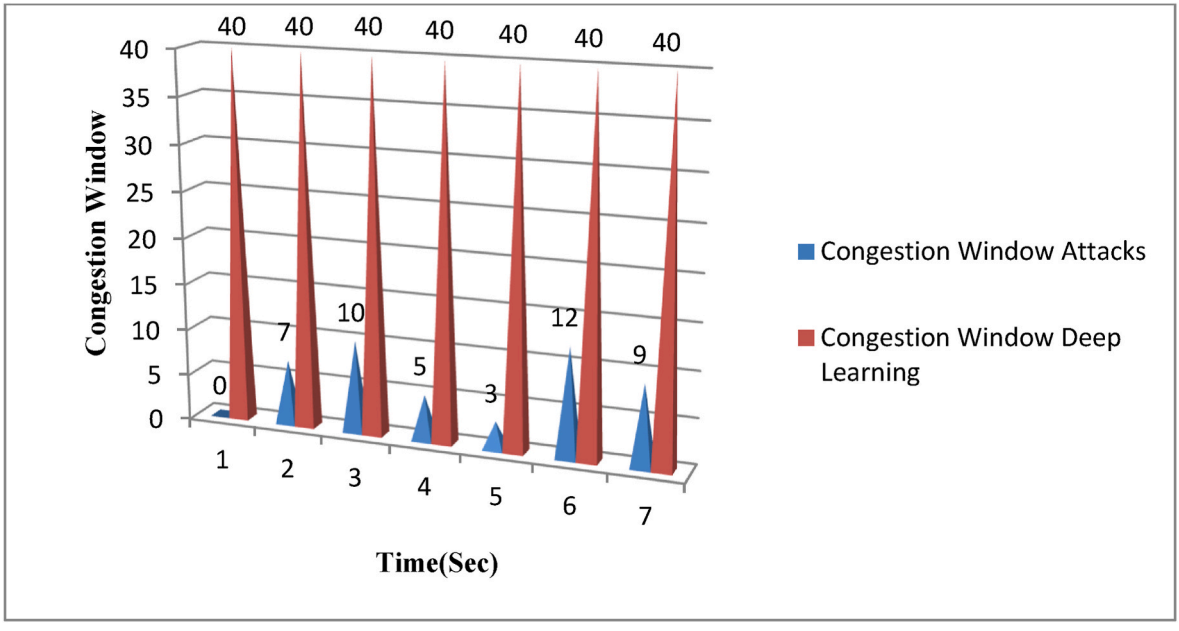

Fig. 6. Congestion changes vs. attacks and deep learning.

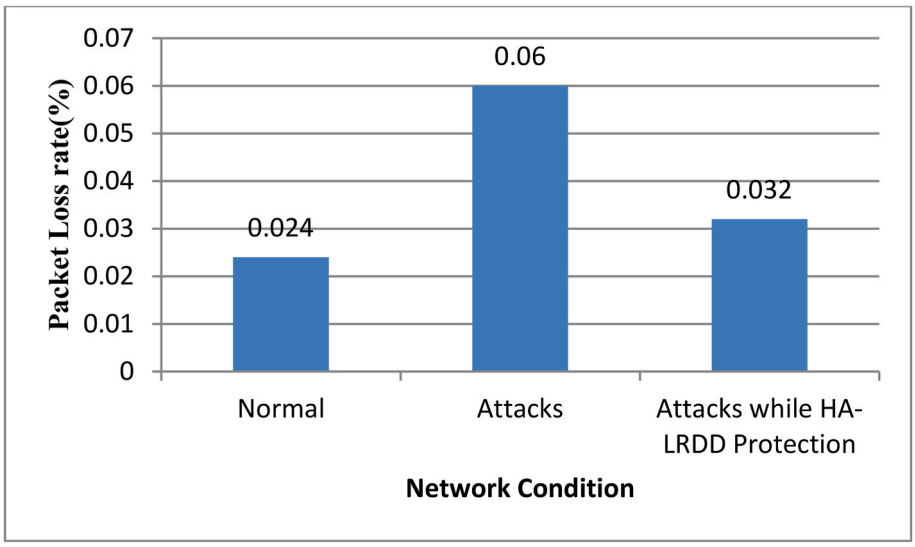

Fig. 7. Packet loss rate analysis under different network conditions.

learning the characteristics of data. The sparse cost function is measured using Eq. (6), while the parameters W and b are updated using Eqs. (7) and (8). The outcome of step 1 is used to initialize the first layer of DNN in step 2 (Fig. 3 (a)). Training parameters are set for the forward propagation algorithm for classification purposes. The cost function is computed using Eq. (5). Then the back propagation algorithm is executed without sparse terms. The back propagation algorithm is executed again to fine-tune the network and adjust weights. In step 3, as shown in Fig. 3 (c), testing is carried out to test network traffic and classify it as benign or malicious.

3.4. Algorithm design: Hybrid Approach for Low-rate DDoS detection (HA-LRDD)

The proposed framework for finding low-rate DDoS attacks uses an algorithm called Hybrid Approach for Low-Rate DDoS Detection (HA-LRDD). This algorithm is used to implement the proposed framework.

Algorithm. Hybrid Approach for Low-Rate DDoS Detection (HA-LRDD)

Algorithm: Hybrid Approach for Low-Rate DDoS Detection (HA-LRDD)

Input: Network traffic data Output: Binary classification of the traffic as normal or DDoS attack

1. **Feature Extraction:**
   - Extract relevant features from the network traffic data (e.g., packet size, packet inter-arrival time, protocol type, etc.) using statistical methods and heuristics.
2. **Normalization:**
   - Normalize the extracted features to ensure that they are on the same scale and do not dominate the analysis.
3. **Deep Neural Networks (DNN):**
   - Train a deep neural network (DNN) on the normalized features to learn the underlying patterns and relationships between them.
   - Use a suitable architecture for the DNN, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), depending on the nature of the features and the task.
4. **Classification:**
   - Classify the network traffic data as normal or DDoS attack using the trained DNN.
   - Use a binary classifier that outputs 1 for DDoS attacks and 0 for normal traffic.
   - Use suitable performance metrics such as accuracy, precision, recall, and F1-score to evaluate the performance of the classifier.
5. **Hybrid Approach:**
   - Combine the above steps in a hybrid approach to achieve better detection accuracy and robustness against different types of DDoS attacks.
   - Use multiple DNNs with different architectures and hyperparameters to learn different aspects of the traffic data and combine their outputs using an ensemble method.
   - Use additional techniques such as unsupervised learning, feature selection, and data augmentation to further improve the performance of the system.
6. **Output:**
   - Output the binary classification results for the network traffic data, indicating whether it is normal or DDoS attack.
   - End.

3.5. Performance metrics

Performance metrics are used to measure the effectiveness of a security system in detecting and preventing intrusions or attacks. There are several metrics that are commonly used to evaluate the performance of a security system, including detection rate (DR) and false positive rate (FPR). Detection rate (DR) is a metric that measures the percentage of detected intrusion data that is correctly identified by the security system. The formula for calculating DR is:

$$\mathrm{DR} = \frac{\text{number of detected intrution data correctly}}{\text{total intrution attack data}} \times 100 \tag{9}$$

False positive rate (FPR) is a metric that measures the percentage of normal data that is incorrectly identified as an intrusion or attack by the security system. The formula for calculating FPR is:

$$\mathrm{FPR} = \frac{\text{miss} - \text{detected rate}}{\text{total normal data}} \times 100 \tag{10}$$

The detection rate and false positive rate metrics defined in Eqs. (9) and (10) are used to evaluate the proposed method and are compared with the state of the art.

4. Experimental results

Experiments are made with the proposed HA-LRDD method, and its performance is compared with different existing methods against varied training data sizes. The existing methods used for evaluation include K-Means, Fuzzy C Means, Surface Learning IDS (SL-IDS), and Incremental Support Vector Machines (ISVM). Observations are made in terms of the training time, detection rate, false positivity rate, congestion window changes due to attacks, and packet loss rate.

This Table 2 (see Table 1). Shows the simulation parameters that were used to set up SAE-based DNN (deep neural network) methods for a certain task.

For the SAE (stacked autoencoder) method, the input data has 2000 nodes, the hidden layer has 600 nodes, and the output layer also has 2000 nodes. The target for sparsity is set to 0.08, which means that the activations of each hidden node should be close to zero for about 8% of the input data samples. The sparse rate (sr) is set to 0.4, which means that only 40% of the hidden nodes will be active at any given time during training. The learning rate (lr) is set to 1, and the dropout rate (dr) is set to 0.3.

For the DNN method, the input data has 2000 nodes, the hidden layer has 600 nodes, and the output layer also has 2000 nodes. The dropout rate (dr) is set to 0.3, and the learning rate (lr) is set to 1.

These parameters were chosen to optimize the performance of the SAE-based DNN methods for the specific task at hand, which could be anything from image recognition to speech processing to natural language understanding.

To aid in comprehension of the proposed SAE-based DNN, all simulation parameters and their respective values have been listed in Table 2, so in all Based methods that use the stochastic gradient descent method for updating model parameters from training sample-based cross entropy errors via layer-wise back-propagation. The simulation of all algorithms performed in the experiments was carried out using MATLAB R2011b software environment.

Table 3 shows the time required for training for various detection algorithms as the size of the training data increases.

Table 3 has five columns representing the number of instances in the training dataset and the corresponding training time (in minutes) for different algorithms - K-Means, FCM, ISVM, SL-IDS, and HA-LRDD. For instance, when the training dataset has one instance, K-Means takes 19.9 min to train, FCM takes 99.5 min, ISVM takes 79.6 min, SL-IDS takes 25.87 min, and HA-LRDD takes 24.875 min. Similarly, for a training dataset with two instances, the training times increase for all algorithms. This trend continues as the size of the training dataset increases.Table 3 provides useful information on the scalability of different detection algorithms as the training dataset size increases. For instance, we can see that HA-LRDD consistently takes the least amount of time to train.

Table 4 displays the performance of different detection algorithms in terms of two metrics: detection rate and false positivity rate. The detection rate represents the percentage of attacks or anomalies that were correctly identified by the algorithm. For instance, the ISVM algorithm detected 93.58% of attacks, while HA-LRDD achieved a higher detection rate of 95.32%. The false positivity rate refers to the percentage of normal or legitimate activities that were mistakenly identified as attacks or anomalies. This is also known as the false alarm rate. For instance, the K-Means algorithm had a false positivity rate of 6.84%, which means that around 7% of normal activities were mistakenly flagged as attacks. On the other hand, the ISVM algorithm had a very low false positivity rate of 0.48%, indicating that it produced very few false alarms. Overall, the results suggest that the HA-LRDD algorithm performed the best in terms of both detection rate and false positivity rate, followed closely by the ISVM and Sl-IDS algorithms. The K-Means and FCM algorithms had lower detection rates and higher false positivity rates, indicating that they were less effective in identifying attacks and more prone to producing false alarms.

As presented in Fig. 5, the performance of different algorithms is compared in terms of detection rate and false positivity rate. The detection algorithms are provided on a horizontal axis. The performance of each detection model is shown on a vertical axis. It was discovered that the algorithms performed differently due to differences in their modes of operation. The proposed HA-LRDD algorithm showed the highest detection rate, at 95.31522%. K-Means has the lowest detection rate of all the algorithms, with 74.99492%. The false-positive rate of the proposed method is 0.56943%, which is better than all other existing methods except ISVM.

As presented in Table 5, it shows the congestion window dynamics in the presence of attacks and in the presence of a deep learning model to handle attacks.

As presented in Fig. 6, the congestion dynamics in the presence of attacks and a deep learning-based detection model are provided. The elapsed time of simulation is presented on the horizontal axis, and the vertical axis shows congestion window details. In the presence of attacks and the proposed detection model, it is analyzed. The results revealed that, in the presence of the attacks (without the proposed solution), the network is always congested. The congestion is minimized when the proposed solution is in place.

As shown in Table 6, the rate of packet loss is measured under different network conditions where tests are run.

As presented in Fig. 7, different network conditions are presented on the horizontal axis, and the packet loss rate is shown on the vertical axis. The packet loss rate is observed as it measures the performance of the network and the effect of a low-rate DDoS attack on the network. Low packet loss rates indicate high performance of the proposed model. Under normal conditions, attack conditions, and attack conditions while the proposed model HA-LRDD is in place, observations are made on packet loss rate. The empirical results revealed that there is the least packet loss rate at 0.024 under normal (no attacks) conditions. When there are attacks, the packet loss is increased by 0.06. In the presence of the attacks and the HA-LRDD, the packet loss rate is reduced to 0.032. It shows the minimal impact on the system even in the presence of attacks when the proposed solution is employed.

5. Conclusion and future work

Considering the contemporary challenge in cloud computing due to DDoS attacks and the evolving strategies of attackers, this paper focuses on building defenses against low-rate DDoS attacks by proposing an AI-enabled framework. The rationale behind this is that low-rate DDoS attacks are one of the strategies of attackers to disrupt cloud services.

Container-based technology enables cloud computing to have light-weight approaches to resource utilization and flexibility in scaling services. As cloud infrastructure is evolving rapidly, it is essential to protect it from such attacks. From the review of the literature, it is understood that the existing methods lack adequate mechanisms to deal with low-rate DDoS attacks. Based on the need for detecting and defeating such attacks, we proposed a framework named the Low-Rate DDoS Attack Detection Framework (LRDADF). Since low-rate DDoS attacks are difficult to defeat, we proposed a mathematical model to realize a mitigation strategy besides employing deep learning methods to have an effective means of detecting such attacks. We proposed an algorithm named Hybrid Approach for Low-Rate DDoS Detection (HA-LRDD). The algorithm uses artificial intelligence (AI)-enabled methods comprising a deep convolutional neural network (CNN) and a deep autoencoder. We defined another algorithm named Dynamic Low-Rate DDoS Mitigation (DLDM), which reduces the impact of the attack once it is detected. It also tries to ensure that the attack is defeated and the infrastructure works as usual even under the attack. An extensive simulation study revealed that the proposed framework is able to detect low-rate DDoS attacks and also mitigate the attacks to ensure there is acceptable quality of service in cloud computing environments. In the future, we intend to investigate further deep learning methods to defeat low-rate DDoS attacks more effectively.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

References

[1] Fatemeh Khoda Parast, Chandni Sindhav, Nikam Seema, Izadi Yekta Hadiseh, B. Kent Kenneth, Saqib Hakak, Cloud computing security: a survey of service-based models, Comput. Secur. 114 (2022), 102580.

[2] N. Agrawal, S. Tapaswi, Defense schemes for variants of distributed denial-of-service (DDoS) attacks in cloud computing: a survey, Inf. Secur. J. A Glob. Perspect. 26 (2017) 61–73.

[3] N. Agrawal, S. Tapaswi, Defense mechanisms against DDoS attacks in a cloud computing environment: state-of-the-art and research challenges, IEEE Communications Surveys & Tutorials 21 (2019) 1–27.

[4] Mr K. Karthick, G. Kiruthiga, P. Ms Saraswathi, B. Dhiyanesh, R. Radha, A subset scaling recursive feature collection based DDoS detection using behavioural based ideal neural network for security in A cloud environment, Procedia Comput. Sci. 215 (2022) 509–518.

[5] Q. Yan, F.R. Yu, Q. Gong, J. Li, Software-defned networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: a survey, some research issues, and challenges, IEEE Communications Surveys & Tutorials 18 (2016) 602–622.

[6] A. Shameli-Sendi, M. Mourzandi, M. Fekih-Ahmed, M. Cheriet, Taxonomy of distributed denial of service mitigation approaches for cloud computing, J. Netw. Comput. Appl. 58 (2015) 165–179.

[7] Y. Xiang, K. Li, W. Zhou, Low rate DDoS attack detection and traceback by using new information metrics, IEEE Trans. Inf. Forensics Secur. 6 (2011) 426–437.

[8] N. Agrawal, S. Tapaswi, A lightweight approach to detect the low/high rate IP spoofed cloud DDoS attacks, in: IEEE 7th International Symposium on Cloud and Service Computing (SC2'17), Kanazawa, 2017, pp. 118–123.

[9] J. Luo, X. Yang, J. Wang, J. Xu, J. Sun, K. Long, On a mathematical model for low-rate shrew DDoS, IEEE Trans. Inf. Forensics Secur. 9 (2014) 1069–1083.

[10] P. Sharma, R. Sharma, E.S. Pilli, A.K. Mishra, A detection algorithm for DoS attack in the cloud environment, Proceedings of the 8th Annual ACM India Conference on - Compute 15 (2015) 1–4.

[11] P. Srilakshmi, N. Sujatha, O. Defensive cloud service providers against stealthy denial of service strategy, International Journal of Computer Engineering In Research Trends 3 (2016) 369–375.

[12] N. Agrawal, S. Tapaswi, Defense mechanisms against DDoS attacks in a cloud computing environment: state-of-the-art and research challenges, IEEE Communications Surveys & Tutorials (2019) 1–27.

[13] K. Bhushan, B.B. Gupta, Detecting DDoS attack using software defined network (SDN) in cloud computing environment, in: 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN), 2018, pp. 1–6.

[14] S. MahdaviHezavehi, R. Rahmani, An anomaly-based framework for mitigating effects of DDoS attacks using a third party auditor in cloud computing environments, Cluster Comput. (2020) 1–19.

[15] A.R. Wani, Q.P. Rana, U. Saxena, N. Pandey, Analysis and detection of DDoS attacks on cloud computing environment using machine learning techniques, in: 2019 Amity International Conference on Artificial Intelligence (AICAI), 2019, pp. 1–6.

[16] D. Hu, P. Hong, Y. Chen, FADM: DDoS flooding attack detection and mitigation system in software-defined networking, in: GLOBECOM 2017 - 2017 IEEE Global Communications Conference, 2017, pp. 1–7.

[17] C. Buragohain, N. Medhi, FlowTrApp: an SDN based architecture for DDoS attack detection and mitigation in data centres, in: 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN), 2016, pp. 1–6.

[18] K. Bhushan, B.B. Gupta, Distributed denial of service (DDoS) attack mitigation in software defined network (SDN)-based cloud computing environment, J. Ambient Intell. Hum. Comput. (2018) 1–13.

[19] G.S. Kushwah, V. Ranga, Voting extreme learning machine based distributed denial of service attack detection in cloud computing, J. Inf. Secur. Appl. 53 (2020) 1–12.

[20] A. Sahi, D. Lai, Y. Li, M. Diykh, An efficient DDoS TCP flood attack detection and prevention system in a cloud environment, IEEE Access (2017) 1–13.

[21] S. Dong, R. Jain, K. Abbas, A survey on distributed denial of service (DDoS) attacks in SDN and cloud computing environments, IEEE Access 9 (2019) 1–16.

[22] S.-C. Tsai, I.-H. Liu, C.-T. Lu, C.-H. Chang, J.-S. Li, Defending cloud computing environment against the challenge of DDoS attacks based on software defined network, Smart Innovation, Systems and Technologies 8 (2016) 285–292.

[23] G. Somani, M.S. Gaur, D. Sanghi, M. Conti, M. Rajarajan, R. Buyya, Combating DDoS attacks in the cloud: requirements, trends, and future directions, IEEE Cloud Computing 4 (2017) 22–32.

[24] Z. Liu, X. Yin, H.J. Lee, A new network flow grouping method for preventing periodic shrew DDoS attacks in cloud computing, in: 2016 18th International Conference on Advanced Communication Technology (ICACT), vol. 9, 2016, 1-1.

[25] V. Vidya, K. Padma Kiran, C. Vani, K. Tarakeswar, Two layer encryption be imminent to protected data sharing in cloud computing, International Journal of Computer Engineering In Research Trends 1 (2014) 266–270.

[26] D.J. Prathyusha, G. Kannayaram, A cognitive mechanism for mitigating DDoS attacks using the artificial immune system in a cloud environment, Evolutionary Intelligence 7 (2020) 1–12.

[27] Vinícius de Miranda Rios, Pedro R.M. Inácio, Damien Magoni, Mário Freire, Detection of reductionof-quality DDoS attacks using Fuzzy Logic and machine learning algorithms, in: Computer Networks, vol. 186, Elsevier, 2021, pp. 1–20.

[28] O. Osanaiye, K.-K.R. Choo, M. Dlodlo, Distributed denial of service (DDoS) resilience in cloud: review and conceptual cloud DDoS mitigation framework, J. Netw. Comput. Appl. 67 (2020) 147–165.

[29] J.A. Perez-Diaz, I.A. Valdovinos, K.-K.R. Choo, D. Zhu, A flexible SDN- based architecture for identifying and mitigating low-rate DDoS attacks using machine learning, IEEE Access (2020) 1–15.

[30] N. Agrawal, S. Tapaswi, A lightweight approach to detect the low/high rate IP spoofed cloud DDoS attacks, in: 2017 IEEE 7th International Symposium on Cloud and Service Computing (SC2), 2017, pp. 1–6.

[31] K. Bhushan, B.B. Gupta, Hypothesis test for low-rate DDoS attack detection in cloud computing environment, Procedia Comput. Sci. 132 (2018) 947–955.

[32] Z. Liu, X. Yin, Y. Hu, CPSS LR-DDoS detection and defense in edge computing utilizing DCNN Q-learning, IEEE Access 8 (2020) 42120–42130.

[33] N. Zhang, F. Jaafar, Y. Malik, Low-rate DoS attack detection using PSD based entropy and machine learning, in: 2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), vol. 4, 2019, pp. 1–4.

[34] K.S. Sahoo, D. Puthal, M. Tiwary, J.J.P.C. Rodrigues, B. Sahoo, R. Dash, An early detection of low rate DDoS attack to SDN based data center networks using information distance metrics, Future Generat. Comput. Syst. 89 (2018) 685–697.

[35] P.S.A. A, Optimal virtual machine (VM) load distribution and DDOS attacks detection in cloud computing environment, Journal of Advanced Research in Dynamical and Control Systems 12 (2020) 855–863.

[36] L. Zhou, M. Liao, C. Yuan, H. Zhang, Low-rate DDoS attack detection using expectation of packet size, Secur. Commun. Network. (2017) 1–14.

[37] X. Liu, J. Ren, H. He, Q. Wang, C. Song, Low-rate DDoS attacks detection method using data compression and behavior divergence measurement, Comput. Secur. 100 (2021), 102107.

[38] N. Agrawal, S. Tapaswi, Low rate cloud DDoS attack defense method based on power spectral density analysis, Inf. Process. Lett. 138 (2018) 44–50.

[39] Z. Wu, Q. Pan, M. Yue, L. Liu, Sequence alignment detection of TCP- targeted synchronous low-rate DoS attacks, Comput. Network. 152 (2019) 64–77.