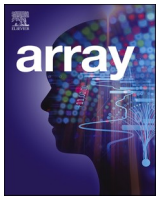
Array 15 (2022) 100229

Contents lists available at ScienceDirect

Array

journal homepage: www.sciencedirect.com/journal/array



Distributed denial of service attack detection in E-government cloud via data clustering

Fargana J. Abdullayeva

Institute of Information Technology, Azerbaijan National Academy of Sciences, 9A, B. Vahabzade Street, AZ1141, Baku, Azerbaijan

ARTICLE INFO

Keywords:
Clustering
Feature selection
DDoS attacks
Cloud security
Network security
Machine learning

ABSTRACT

One of the main essential security issues of cloud computing is the detection and prevention of network intrusions. The gaps in the network directly affect the security of the cloud as it is the foundation of it. Attacks in the cloud are launched either by compromised nodes of the network outside of the cloud or by virtual machines (VMs) within the cloud network. So, monitoring both external and internal traffic of the cloud network is of great importance. In this paper, a machine learning method performing accurate clustering of network data to detect DDoS attacks has been proposed. The method uses a feature selection technique to increase the efficiency of data clustering. To provide the feature selection the PCA algorithm has been used. For the dataset formed on selected features, the DBSCAN (density-based spatial clustering of applications with noise), Agglomerative Clustering, and k-means algorithms are applied. In the experiment, the clustering results of the methods using fewer features were higher on all metrics than the clustering results of the methods using all the features. Compared to the standard algorithms, the PCA + DBSCAN, PCA + Agglomerative, and PCA + k-means algorithms obtained higher values on the Adjusted Rand Index metric and reached 0.8989, 0.9130, 0.9094 values, respectively. The effectiveness of the approach also was evaluated on the other clustering metrics and obtained high results. The proposed system can be installed in both internal and external cloud infrastructure. This allows, to detect attacks on the external cloud network, as well as on the internal physical network or in the virtual network between hypervisors.

1. Introduction

E-government is an innovative transformation of technology, the state can use this system to improve citizen-state relations by providing quality services to citizens. Cloud computing is a new way to provide services over the Internet. In cloud computing, all resources (hardware, software, network resources) are provided as a service. Cloud-based e-government is considered to be the leading paradigm of distributed computing systems in the field of e-government. Cloud-based e-government uses cloud services as a fundamental element in building intelligent networks of collaborative applications distributed within or outside the state.

Data scaling; auditing and logging; rolling out new instances, replication, migration; disaster detection; performance and scaling; reporting and intelligence; policy management; system integration and legacy programs; migration to new technology; ease of application; cost savings; scalability; accessibility from any location; green computing system are the benefits of cloud computing for e-government. The cloud-based e-government system has many advantages for the state. These benefits include cost reduction, distributed storage, affordable access to resources, scalability, and accountability. Despite the advantages of cloud-based e-government, this technology is exposed to numerous security threats.

Cloud security policy is governed by the CIA triad model of confidentiality, integrity, and accessibility [1]. Attackers try to violate the cloud security policy by carrying out various interventions, using gaps in the applied protocols.

NIST (National Institute of Standards and Technology) defines the intrusion as a threat to security policy (confidentiality, integrity, and accessibility) or as avoidance of the computer and network security mechanisms.

The CSA (Cloud Security Alliance) has identified several types of threats to the cloud: Denial-of-Service (DoS), data breach, unauthorized access, insecure APIs, vulnerable applications, account hacking, malicious content, data loss, and abuse of services [2]. One of the most common types of attacks that cause serious damage to the cloud and affect its effectiveness is DoS. DoS is a method used by hackers to disrupt the availability of services and resources to its legitimate users.

E-mail address: a_farqana@mail.ru.

https://doi.org/10.1016/j.array.2022.100229
Received 22 December 2021; Received in revised form 6 July 2022; Accepted 6 July 2022
Available online 14 July 2022

Distributed DoS (DDoS) uses several systems called botnets or zombies to attack the target cloud network [3]. Features of DoS attacks are poor network performance, very slow response from a website or application, violation of the availability of cloud services, a sudden increase in bandwidth consumption, sudden increase in server resource usage, instability of network connection, incorrect response, refusal of service requests.

Since DDoS attacks are highly adaptable, so they can generate traffic according to the nature of the target object. Due to these features, the DDoS attack covers not only traditional networks but also a network of higher technologies.

Since DDoS attacks can occur in different layers of the Internet, including the cloud, methods of countering attacks must be designed and used in separate layers [4,35]. The working principle of the protection mechanism of each layer is different. For this reason, the paper proposed a method to detect DDoS attacks in the network layer.

In Imperva's global 2019 report on DDoS threats in May December, 3643 network layer DDoS attacks were recorded [5]. According to the Kaspersky report, in the 4th quarter of 2019, the number of DDoS attacks doubled [6]. The report [7] notes that the number of attacks on SaaS services has tripled in 2018 and increased to 41% in 2018 from 13% in 2017. Attacks on data centers and cloud services have also risen from 11% to 34% in these years.

The scale of DDoS attacks is gradually expanding. Over the last decade, DDoS attack traffic has increased from 70 Gbps to 1.35 Tbps (Terabytes per second, Tbps) [8]. Attackers use cloud computing and IoT (Internet of Things) to generate large amounts of attack traffic [9].

LOIC (Low Orbit Ion Canon), XOIC, DDoSIM, DAVOSET, and PyLoris are tools used by attackers to perform various DDoS attacks. HULK, R-U-Dead-Yet, and GoldenEye HTTP DoS are tools used to create various DDoS attacks targeting cloud services.

DDoS attacks are usually detected by classifying network traffic into genuine and malicious classes. These methods are divided into signature-based, anomaly-based, and hybrid categories. [10] proposes a machine learning approach based on the C.4.5 algorithm for detecting DDoS attacks. The detection of DDoS attacks in the approach is based on signatures. [11] analyses the existing approaches to detect DDoS attacks in the cloud and compares their detection accuracy. [12] explores the possibility of combining cloud computing and SDN technologies to protect against DDoS attacks. An architecture called DaMask is proposed to prevent DDoS attacks using SDN. The DaMask model consists of 2 modules: an anomaly-based attack detection module called as DaMask-D, and an attack prevention module called DaMask-M. In the DaMask-D module, anomalies are detected based on a graphical probabilistic inference model. [13] proposed a method for detection of Low-rate DDoS (LDDoS) attacks based on frequency. The approach is based on the power spectral density (PSD) algorithm and detects an attack by analysing real-time traffic. The approach experimentally is applied in the OpenStack real cloud environment.

These methods extremely depend on the correct labeling of the data, namely on the distinguishing of the attack classes. As the labeling is done autonomously, so this procedure limits the real-time application of the algorithm. In this paper, the data clustering approach is used to overcome this problem. Clustering does not require labeling the data by user and significantly reduces the search space. For the detection of DDoS attacks, the proposed approach uses feature selection to provide accurate clustering of the network data. In this paper, the PCA algorithm is used to select the features. DBSCAN, Agglomerative Clustering, and k-means algorithms are applied to the dataset formed on the basis of selected features.

Feeding raw data or a large number of features into the input of the clustering algorithms is not considered effective. Here, the purpose of feature selection usage is to remove redundant features, choose the most important features from them, and reduce the search space.

Also, in the existing works, the DDoS attacks were detected using network traffic parameters, such as IP address, TCP flags, etc. separately [14]. Web traffic-specific features, such as IP addresses and TCP flags, contain low information to detect such types of attacks. Only the use of these features can hide the real state of the packages [15]. For example, IP address falsification or high-frequently changing can occur using various methods (fast-flux). In the proposed approach, the network traffic features are fed into the input of the algorithm in a vector form. This allows participation of not only one but also several features at the same time in the detection process of DDoS attacks. This increases the probability of a DDoS attack accurately being detected.

All these problems show that cloud IDS (Intrusion Detection System) needs to analyse large amounts of network traffic, effectively detect new types of attacks, and achieve high detection accuracy by making fewer errors.

The scientific contributions of the paper are the following:

1. For the detection of DDoS attacks, a hybrid model that combines Principle Component Analysis and clustering algorithms has been constructed. The proposed approach is evaluated in terms of the clustering quality, against the baseline algorithms such as Agglomerative Clustering, k-means, and DBSCAN. On the CSE-CIC-IDS2018, NSL-KDD, and HTTP CSIC 2010 datasets the experiments have been provided.
2. By using feature selection, the redundant features are removed, and the baseline feature set is reduced.

The paper is constructed as follows. Section 2 presents an overview of the related work. Section 3 explains the Cloud network infrastructure, introduces the DDoS attack scenario on the Cloud, and presents the framework of the proposed DDoS attack detection system model constructed in this paper. Section 4 describes machine learning methods that are used in the proposed hybrid model. Section 5 gives the clustering metrics used to evaluate the quality of the clustering. Section 6 presents the application scenario of the proposed system in a cloud network. Section 7 provides the results obtained from the experiments on CSE-CIC-IDS2018, NSL-KDD, and HTTP CSIC 2010 datasets. Section 8 gives the conclusion of the research.

2. Related work

The difference between traditional security systems and cloud security systems is that the cloud security systems are installed and launched on separate components of the cloud. For example, security mechanisms installed in VMs monitor the security status of VMs. Security mechanisms installed in the hypervisor monitor the condition of the hypervisor. Security mechanisms installed at the nodes of the cloud network monitor the security of the cloud network [21].

Numerous studies were conducted to detect DDoS attacks in the cloud. Supervised and unsupervised machine learning algorithms are widely used in the detection of DDoS attacks. While the goal of supervised approaches is to conduct research on pre-labeled benchmark data, the goal of unsupervised approaches is to label data by referring them to separate spaces according to similarity criteria. One of the most common unsupervised approaches is clustering algorithms. Clustered data can be used to label the data points.

In [16] a semi-supervised machine learning method has been proposed. Firstly, the network traffic is divided into the desired number of classes by the application of the unsupervised method, then the clustered data is labeled as normal, DDoS, and malicious traffic classes by using the voting method, at last, the detection of unknown traffic classes is provided by using the supervised approach. In the unsupervised learning phase of the proposed approach, to map the data into the low dimensional space, by applying the PCA method the feature extraction is performed. Although the work addresses the recent DDoS vectors, the performance of the method has been evaluated only by entropy.

[17] addressed the issue of detecting DDoS attacks carried out by insiders. Traditional defense mechanisms, such as firewalls, cannot

detect attacks launched by insiders. In this study, an approach to detecting abnormal intrusions at the hypervisor level has been proposed to prevent DDoS attacks between virtual machines. An approach uses an evolutionary neural network. The evolutionary neural network is created based on the integration of the PSO and the Neural Network, and the main aim of this model is to classify and detect the traffic flow between virtual machines. The drawback of this method is that it is computationally expensive.

In [18], for the detection of DDoS attacks in the clouds, a multistage anomaly detection method has been proposed. DDoS attacks targeting cloud web services were categorized into oversized payload, coercive parsing, and flooding attacks. In Ref. [10], to detect DDoS attacks, a machine learning approach based on the C.4.5 algorithm has been proposed. In this approach, the DDoS attack detection is based on signatures. The limitation of this approach is the detection of known attacks that attack behavior is already evident. [19] proposed the Confidence-Based Filtering method (CBF) to detect DDoS attacks in the cloud computing environment. This method uses two concepts: one named confidence for measuring correlation patterns, and another named CBF (Confidence-Based Filtering) score for judging the legitimacy of packets. The shortcoming of this method is based on assigning a threshold value to detect DDoS attacks.

[12] investigated the possibility of combining cloud computing and SDN technologies to protect against DDoS attacks. In this study, the DDoS attack prevention architecture constructed on SDN is called DaMask. The DaMask model consists of 2 modules: an anomaly-based attack detection module, called as DaMask-D, and an attack prevention module, called DaMask-M. In the DaMask-D module, anomalies are detected based on a graphical probabilistic inference model. The disadvantage of this approach is that it does not take into account the possibility of false positives occurring. Blocking of all suspected traffic in an instant in some cases results in the disruption of real user services.

[20] addressed the DDoS attack detection in an infrastructure cloud. In the infrastructure cloud, the DDoS attack typically targets the server's core resources such as CPU, memory, disk space, bandwidth, TCP connection, open files, and so on. It is experimentally proven, that along with the target object, the stakeholders of the cloud are also exposed to DDoS attacks. The requirements that mitigate the consequences of DDoS in the cloud have been determined.

[1] analysed the DDoS attack detection methods in the cloud. The taxonomy and a conceptual model of the preventive mechanism against DDoS attacks in the cloud are given. Here, the current research issues and future research directions have been investigated. Common DDoS attacks targeting cloud computing were analysed and categorized into two groups: application layer attacks and infrastructure layer attacks.

[22] proposed a distributed network security approach at the hypervisor level. The developed system is implemented in each processing server of cloud computing. To detect intrusions, the security system on each server monitors network traffic coming in and out of the virtual network, internal network, and external network associated with the appropriate virtual machine. From the cloud network traffic, the features are obtained using the Binary Bat Algorithm (BBA). Obtained features fed into the input of the Random Forest Algorithm and in this way, the detection of intrusions from the cloud network traffic and alert generation are provided. The effectiveness of the proposed approach is evaluated on the UNSW-NB15 and CICIDS-2017 datasets. A drawback of the proposed method is that it produces high false alarm rates and cannot accurately define the baseline of a normal profile.

[23] proposed MLP-based method of optimal feature selection for the detection of DDoS attacks. Due to the employment of a supervised method, the proposed approach in this study cannot detect unknown attacks.

In [24], for the network anomalies IDS, an intellectual approach based on deep neural networks is developed. The proposed IGASAA hybrid optimization approach uses the Genetic Algorithm and the Simulated Annealing Algorithm (SAA). The IDS is called MLIDS–Machine Learning based Intrusion Detection System. Here, the genetic algorithm is developed by applying parallel processing and fitness ratio hashing strategies. As a result of the algorithm development, the processing time and solution convergence time are minimized, and the computing power is saved. To optimize the heuristic search the SAA algorithm is applied to the Improved Genetic Algorithm (IGA). To evaluate the effectiveness of the method the CloudSim 4.0 simulator and the CICIDS2017, NSL-KDD (2015 version), and CIDDS-001 datasets open for scientific research are used. The proposed system can be installed in both internal and external cloud infrastructure. This allows detecting attacks on an external cloud network as well as on an internal physical network or in a virtual network created between hypervisors. One limitation of this approach is its failure to detect attacks that are not predefined in the system.

In [25] for the detection of DDoS attacks in the network layer, DBSCAN, and PCA-based classification method has been proposed. The difference in this research is that here the problem of clustering is turned into the classification issue, and the effectiveness of the method is evaluated based on classification metrics. The classification issue is considered as a static approach. This approach is considered unsuitable for dynamic systems such as the cloud, in which the users get access and leave the system instantly. Besides, the approach proposed in our study, in contrast to the existing one, can detect new types of attacks of unknown nature due to the clustering methodology. The application of the dimensionality reduction technique in the approach allows the detection method to work quickly and efficiently on large amounts of data.

In [26], the network traffic clustering method based on the DBSCAN algorithm for the detection of DDoS attacks has been proposed. The proposed approach consists of three phases: the analysis phase, the clustering phase, and the prevention phase. The disadvantage of this study is the failure to provide the selection of informative features that significantly increase the efficiency of clustering.

In [27], to detect DDoS attacks in the network environment the hybrid approach based on a combination of Taguchi and deep autoencoder methods has been proposed. Note that the training of deep learning methods requires a high volume of labeled data. However, due to the lack of availability of large amounts of data labeled with attack classes, the prediction of these attacks via deep learning on a small amount of data leads to incorrect results. Machine learning methods, which differ from Deep Learning methods, can perform effectively with small amounts of data too.

In [28], an approach for the detection of application-layer DDoS attacks is proposed. Firstly, a dataset containing nine features has been created to determine the type of user requests to the applications. In this study, a potential user behavior patterns profile was created, and deviations from these patterns were considered as an anomaly. Various machine learning methods have been applied to the created dataset to detect attacks. The difference in this research is that the k-means, DBSCAN, and PCA has been used as classification algorithms. The data clustering problems were not considered in this study.

In [29], a DDoS detection system based on the artificial neural network called a voting extreme learning machine (V-ELM) has been proposed. To implement this process, several Extreme learning machines were used simultaneously in the V-ELM classifier. The results of all these machines are combined using a majority vote technique to get the final result.

One of the attacks that threaten IT services of cloud computing is IoT-based distributed denial of service attacks. Currently, the number of IoT devices with security holes is increasing rapidly. Bot writers take control of these types of vulnerable IoT devices and launch massive DDoS attacks against cloud systems. In Ref. [30], a new approach based on gesture verification was proposed to implement effective prevention of DDoS attacks. Here, obstacles to customers are generated based on gesture verification. IoT devices do not have the ability to draw gestures to overcome the barriers. Gestures here can be just drawing a circle or a line.

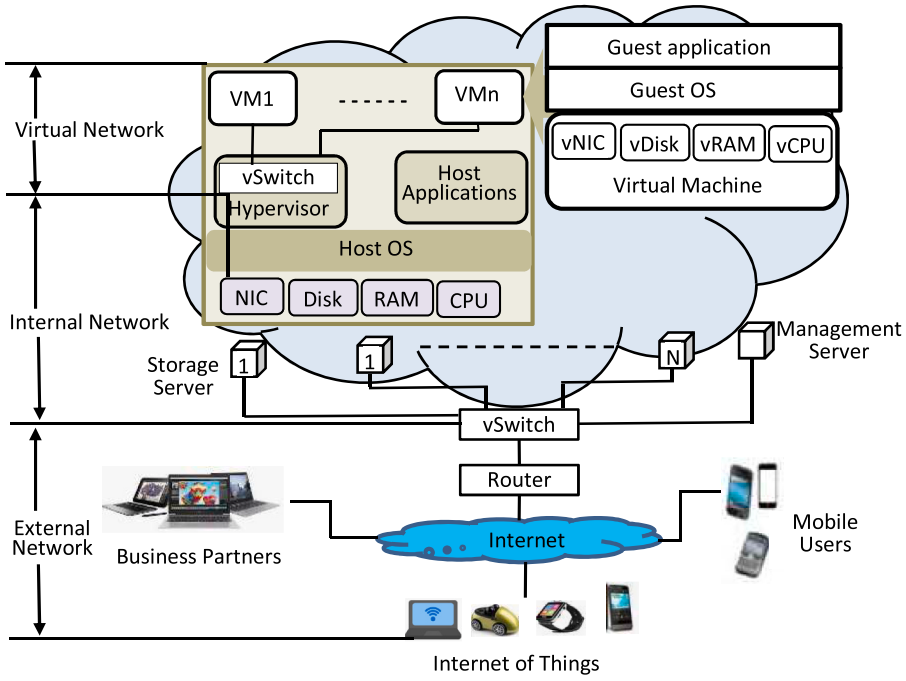


Fig. 1. Cloud network infrastructure.

[31] proposed a system called LSTM-CLOUD based on the Long Short-Term Memory (LSTM) method for the detection of DDoS attacks in an open cloud network environment. The method detects DDoS attacks based on the signature. LSTM-CLOUD model consists of two blocks: detection and protection. In the first module, attacks are detected by applying LSTM, and in the second module, if the attacks are detected, the defense mechanism is activated to defend the cloud system.

In [32], an improved self-adaptive evolutionary extreme learning machine method (SaE-ELM) was proposed. This model is improved by incorporating two more features. Firstly, it can adapt the best suitable crossover operator. Secondly, it can automatically determine the appropriate number of hidden layer neurons. These features improve the learning and classification capabilities of the model.

[33] proposed an efficient DoS attack detection system that uses the Oppositional Crow Search Algorithm (OCSA), which integrates the Crow Search Algorithm (CSA) and Opposition Based Learning (OBL) method to address such types of issues. The proposed system consists of two stages: selection of features using OCSA and classification using Recurrent Neural Network (RNN). In this model, the essential features are selected using the OCSA algorithm and fed to the input of the RNN classifier, and classification of the data is provided.

Recently, the number of application-layer DDOS attacks targeting cloud web services has increased [34]. The goal of these attacks is to gain access to resources by sending SOAP (Simple Object Access Protocol) requests with malicious XML content. While these requests look like real packets their detection at the network and transport layer (TCP/IP) is not possible. [35] proposed an intelligent, fast, and adaptive approach for detecting XML and HTTP type application-layer DDoS attacks. The intelligent system works by extracting several features and using them to construct a model for typical requests. Finally, outlier detection is used and detection of malicious requests is provided.

[36] proposed a deep learning-based model for DDoS attack detection. The proposed model has three main steps. At first, the data is fed to the DNN Input Layer, then into the first hidden layer. This layer like other hidden layers in this model is activated using Rectified Linear Unit (ReLU) activation function and the output from this layer is passed into three more dense layers all of which are activated using ReLU separately. For countering overfitting dropout is applied to each of the three

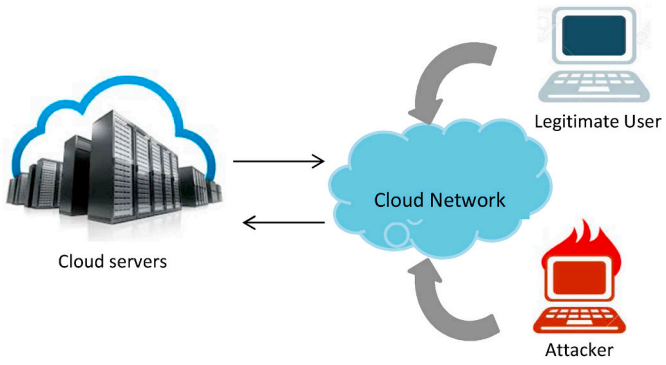


Fig. 2. Cloud network interaction.

layers, the outputs are concatenated in the concatenate layer then fed as input to the fourth and last layer then into the output layer which is activated using a sigmoid function.

[37] analysed deep learning models including restricted Boltzmann machines, deep belief networks, and deep autoencoders for DDoS attack detection. The performance of each deep learning model was evaluated using two new real traffic datasets, namely, the CSE-CIC-IDS2018 dataset and the Bot-IoT dataset. The comparison of deep learning models with machine learning models is provided.

3. Cloud network infrastructure and DDoS attack scenario

One of the important features of cloud computing is network virtualization. In cloud computing, network virtualization allows separate virtual networks to work on a common infrastructure. The cloud network infrastructure consists of three different networks (Fig. 1): virtual network, internal network, and external network.

The virtual network creates communication between VMs on the same physical server. Through the internal network, various components such as cloud management systems, storage systems, network servers, etc. can communicate with each other. The external network acts as the main interface between the cloud user (front end) and the

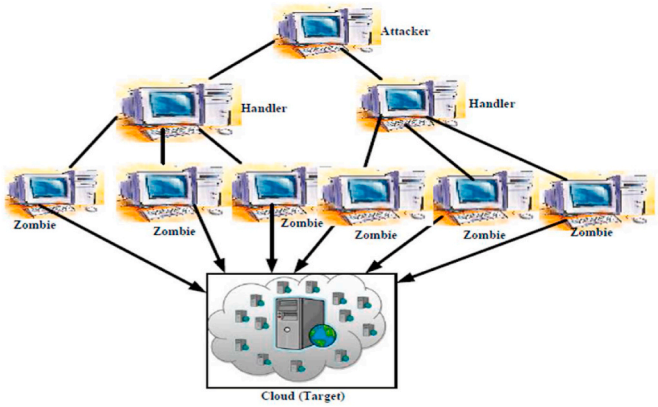


Fig. 3. DDoS attack in the cloud [1,40]

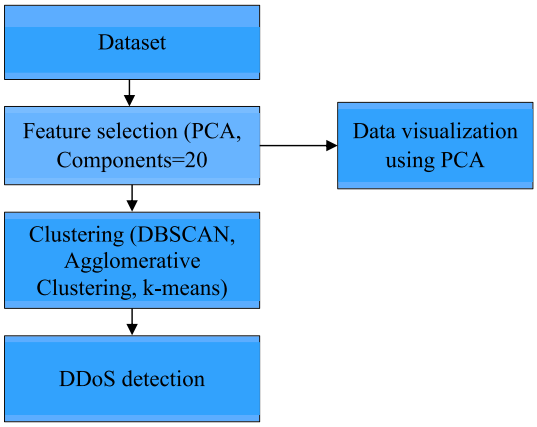


Fig. 4. DDoS attacks detection system.

cloud service provider (back end). All these networks provide the successful delivery of cloud services to users.

Network intrusions are the major threat to the cloud, by which the disruption of service availability, cloud bandwidth, resources, and applications are exposed to attack influence [38]. In the DDoS attack, several botnets provide attacks by sending a large number of requests to the target server. The interaction scenario of real users and attackers with the cloud network is illustrated in Fig. 2.

In Fig. 2, the parties send requests to the cloud server to establish a connection.

In a DDoS scenario on Cloud, an attacker sends a command to the machine layer called handlers to carry out an attack (Fig. 3). Handlers scan vulnerable servers and hosts on the Internet and to control these vulnerable machines install malware. Infected (compromised) machines are called zombies, and the whole network is called a botnet. The botnet is a network of compromised machines (computers) operated by attackers to carry out distributed attacks [39]. Zombies in the botnet are used to attack the target directly and create the Denial of Service. Because of the distributed architecture of the botnet, which consists of a large number of compromised machines used to attack a target object, this attack is called the Distributed Denial of Service attack. Here, the compromised nodes are managed by the Command & Control (C&C) server, controlled by the attacker. Zombies can also collect information from the target object and transfer this information back to the controllers to create feedback between C&C and the attacker.

The general operation of the DDoS attack is illustrated in Fig. 3.

During the DDoS attack, several, sometimes thousands of IP addresses are used to initiate the cyber-attack. And this makes it difficult to prevent DDoS attacks [41].

DDoS attacks on the Internet target specific layers of the OSI model: *Application layer attacks* (target layer 7) and *protocol layer attacks* (target layers 3 and 4).

There are two ways to carry out DDoS attacks on the Internet:

1) *Network/transport layer attacks.* Breaks the legitimate user's connection by consuming bandwidth, router performance, or network resources. Attackers use protocols in the network and transport layers (OSI Layers 3 and 4) to attack the target host. Examples of such attacks are TCP Syn flood, UDP flood, and ICMP flood attacks.
2) *Application layer attacks.* Disables services to legitimate users by consuming server resources (e.g., sockets, CPU, memory, disk/database bandwidth, and input/output (I/O) bandwidth). The attacker uses system and protocol vulnerabilities to disrupt the availability of cloud resources. Examples of application level attacks are HTTP attacks.

In addition to the above attacks, the infrastructure attack can also occur in the clouds. These attacks target the components such as cloud storage, network bandwidth, CPU, and TCP buffer, and make them inaccessible to real cloud users. On the infrastructure layer, the DDoS attacks use only the IP address of the target object. Here exploitation of any vulnerability is not carried out.

In the OSI model, HTTP is built on top of TCP. TCP is used to establish communication between two machines, and HTTP uses this connection to transfer data between the server and the client. HTTP cannot perform without TCP.

To prevent the Cloud from DDoS attacks a detection system is proposed. The proposed DDoS detection system architecture is based on a synthesis of PCA and clustering algorithms and is illustrated in Fig. 4.

Firstly, we extract the most important features by applying the PCA algorithm to the data. Then DDoS attacks are detected by applying clustering algorithms to the dataset, formed on the basis of the selected features.

The workflow of the DDoS attack detection approach consists of the following steps:

Step 1. Input the X data matrix;

Step 2. Find covariance matrix YY^T, where Y represents the centered data matrix $Y = (y_1,\ y_2,\ \ldots,\ y_n)$, Y^T is the transpose of Y;

Step 3. Select the p number of key components with the highest variations;

Step 4. Create a conversion matrix W consisting of a p number of principal components;

Step 5. Find the reduced dataset D;

Step 6. Divide the dataset D into k number of clusters using clustering algorithms.

4. Machine learning methods

Principal Component Analysis (PCA). PCA–is a multivariate method used to analyse a specific data matrix, in which observations are described by interrelated dependent variables. The main goal of the method is to extract the most important features from the data matrix and describe them as a certain set of new orthogonal variables called principal components. Here, the main components can be obtained by Singular Value Decomposition (SVD) or calculation of the eigenvectors. The Singular Value Decomposition of the X matrix is calculated by the following equation:

$$X = P\Delta Q^T \tag{1}$$

where P is the $I \times L$ dimensional matrix of left singular vectors, Q is the $J \times L$ dimensional matrix of right singular vectors, is the diagonal matrix of singular values.

4.1. K-means clustering

The main goal of the K-means clustering algorithm is to divide the n number of different observations into a k number of clusters. The workflow of the k-means clustering algorithm consists of the following steps:

Step 1. Random selection of the k number of cluster centers;

Step 2. Assign each i-th data point x_i to the nearest cluster center using the Euclidean distance (d). Assume that a finite point set $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^m$ is given. The hard clustering problem is the distribution of the points of the set X into a given number k of nonempty, pairwise disjoint subsets $C_j \subseteq X$, $j = 1, \ldots, k$ such that $X = \cup_{j=1}^{k} C_j$. The sets C_j are called clusters, $C = \{C_1, \ldots, C_k\}$. The cluster C_j is identified by its center, $c_j \in \mathbb{R}^m$, $j = 1, \ldots, k$.

$$KM(X, C) = \sum_{i=1}^{n} \min_{j=1,\ldots,k} d\left(x_i, c_j\right) \tag{2}$$

$$d(p, q) = \sqrt{\sum_{i=1}^{m} (p_i - q_i)^2} \tag{3}$$

where x_i is a i-th data point, c_j is a j-th cluster center, p and q are pair of samples in an n-dimensional feature space.

Step 3. Each cluster center is updated periodically, in the form of the average value of all points assigned to the cluster.

Step 4. Steps 2–4 are continued until the cluster centers stabilize. The process is stopped after achieving stability.

4.2. DBSCAN

DBSCAN is a density-based clustering technique. The steps of the DBSCAN clustering algorithm are as follows:

Let $X = \{x_1, x_2, x_3, \ldots, x_n\}$ is the samples of the dataset. In DBSCAN two parameters are used: ε (eps) and the minimum number of samples needed to construct a cluster (minPts). Epsilon (ε) refers to the maximum distance between two points to be said that they are in the same cluster.

Step 1. Beginning with the random starting point that is not visited.

Step 2. Using ε select the neighborhood of this point. All points at a distance ε are neighborhood points. In the experiments, the parameters chosen for eps and minPts are 0.0375 and 50, respectively.

Step 3. If there exists a sufficient neighborhood around this point then the clustering process starts and the point is marked as visited or this point is labeled as noise. Then that point is considered as the point included in the cluster.

Step 4. If the point is defined as part of a cluster then its ε neighborhood becomes a part of a cluster too. Above mentioned steps continue to repeat to all ε neighborhood points. This process is repeated till every point in the cluster is defined.

Step 5. A new unvisited point is selected, guiding to the finding of the next cluster or noise.

Step 6. Above mentioned steps go on until whole points are noted as visited.

5. Clustering metrics

To evaluate the effectiveness of the clustering methods Homogeneity score, Completeness score, V-measure, Adjusted Rand Index, Adjusted Mutual Information, and Silhouette score metrics are used.

Homogeneity score. Each cluster contains only members of a single class. The homogeneity score is formally given by:

$$h = 1 - \frac{H(C|K)}{H(C)} \tag{4}$$

where $H(C|K)$ is the conditional entropy of the classes C given the cluster assignment K, $H(C)$ is the entropy of the classes C.

Completeness score. All members of a given class are assigned to the same cluster. The completeness score is formally given by:

$$c = 1 - \frac{H(K|C)}{H(K)} \tag{5}$$

V-measure is used for comparing the clustering results with the real class labels of data points or with the distinct clustering. It is determined as the harmonic mean of homogeneity (h) and completeness (c) of the clustering:

$$V_\beta = (1 + \beta)\frac{h \cdot c}{\beta \cdot h + c} \tag{6}$$

From the information theory h and c can be expressed in terms of the mutual information and entropy measures. Homogeneity (h) is maximized when each cluster contains elements of as few different classes as possible. Completeness (c) aims to put all elements of each class in single clusters. The β parameter ($\beta > 0$) could be used to control the weights of h and c in the final measure. If $\beta > 1$, completeness has more weight, and when $\beta < 1$ it's homogeneity.

Adjusted Rand Index (ARI) is a measure of the similarity between the two data clustering. From a mathematical standpoint, the Rand index is related to the prediction accuracy but is applicable even when the original class labels are not used. Rand Index is a metric that calculates a similarity rate between two clusterings. For this calculation, the rand index considers all pairs of samples and counts pairs that are assigned in similar or different clusters in the predicted and true clustering. Afterward, the raw Rand Index score is 'adjusted for chance' into the Adjusted Rand Index score by using the following formula:

$$AdjustedRI = \frac{(RI - ExpectedRI)}{(\max(RI) - ExpectedRI)} \tag{7}$$

RI is the Rand index and it can be written as $RI = \frac{a+b}{C_2^{n_{samples}}}$, where a is the number of pairs that are in the same set in both clustering, b is the number of pairs that are in different sets in both clusterings, $C_2^{n_{samples}}$ is the number of possible pairs in the dataset.

Adjusted Mutual Information. Mutual Information is a metric that measures the consensus of two assignments, ignoring permutations. Two different normalized versions of this measure are available, Normalized Mutual Information (NMI) and Adjusted Mutual Information (AMI). The adjusted mutual information can be calculated using a similar form to that of the adjusted Rand index:

$$AMI = \frac{MI - E[MI]}{mean(H(U),\ H(V)) - E(MI)} \tag{8}$$

Silhouette score is a method for evaluating the quality of clustering. Particularly, it provides a quantitative way to measure how well each point lies within its cluster in comparison to the other clusters. The Silhouette value for the i-th data point is:

$$s_i = \frac{b_i - a_i}{\max(a_i,\ b_i)} \tag{9}$$

where a_i is the average distance from the i-th point to the other points in the same cluster z_i, $b_i \stackrel{def}{=} \min_{k \neq z_i} b_{ik}$, where b_{ik} is the average distance from the i-th point to the points in the k-th cluster. Note that $-1 \leq s_j \leq 1$, and that s_j is close to 1 when the i -th point lies well within its own cluster. Higher values indicate better separation of clusters (point distances).

6. Application scenario of the proposed system to the cloud network

The goal of the developed system is to detect attacks from both inside and outside of the cloud environment by monitoring network traffic while maintaining the confidentiality, accessibility, completeness, and

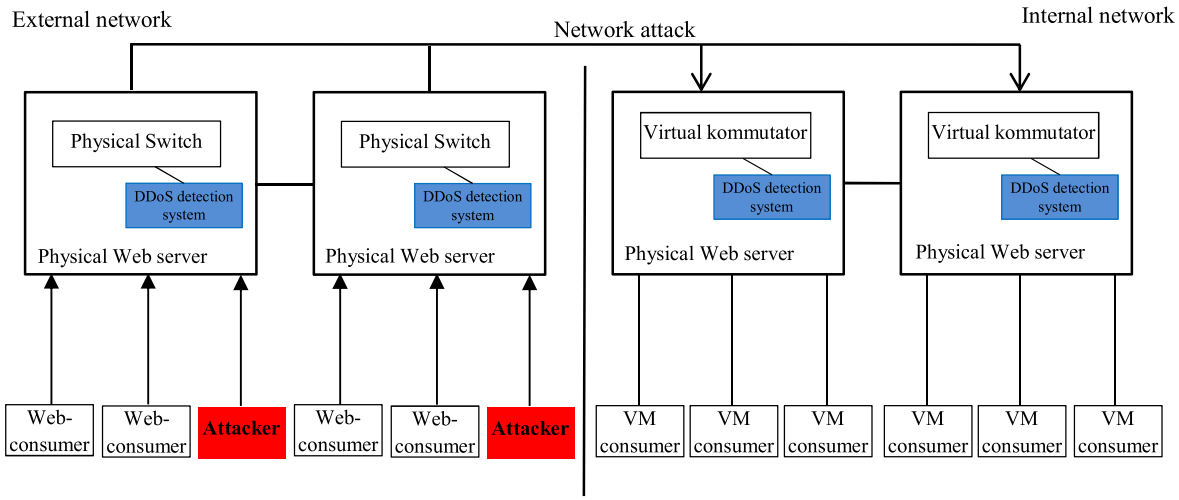


Fig. 5. Application of DDoS detection system to the cloud environment.

performance of the cloud resources and provided services.

The internal network of the cloud infrastructure is a virtual environment. In the virtual environment, there exist many virtual machines on the same physical server. These virtual machines communicate with each other via a virtual switch without leaving the physical server. Because they interact within a physical server, security systems installed on the local network cannot see this network traffic. If the traffic flow between these VMs does not pass through the security mechanism, this situation creates a gap for all types of attacks. Here, the attacker compromises one of the VMs and uses it to gain control over other VMs within the same hypervisor. The control of this breaking operation of the attacker is impossible. Besides, the virtual environment is exposed to various threats, mainly concentrated in the hypervisor: Hyper jacking, VM escape, VM migration, VM theft, and Inter-VM traffic.

The constructed DDoS attack detection system in this study is deployed in the cloud's internal and external networks (Fig. 5).

The developed system is implemented on each processing server of the cloud. For the detection of the DDoS attack, the security system installed on each server monitors network traffic coming from the virtual network, internal network, and external network related to the appropriate virtual machine.

External network. A DDoS detection system placed on the outside of the cloud detects network attacks implemented by compromised nodes of the external cloud network or attackers connected to the Internet. The goal of these attackers is to gain access to the internal cloud by bypassing the firewall. Here, the DDoS detection system acts as a secondary protection layer after the firewall to eliminate its vulnerabilities.

Internal network. Deployment of a DDoS attack detection system (sensor) on virtual servers within the cloud, allows the detection of attacks occurring in the internal network of the cloud. The DDoS detection system installed here monitors both virtual traffic and traffic flow transferred from the virtual server to the physical server. It is not advisable to install a security system on every VM, as this will create an additional workload and complicate the work of the VM. Besides, the migration, provisioning, and de-provisioning of VMs are implemented dynamically so the installation of separate security mechanisms on each VMs also poses management challenges.

7. Experiments

The testing process of the DDoS attack detection model was run at the AzScienceNet Data Center of the Institute of Information Technology

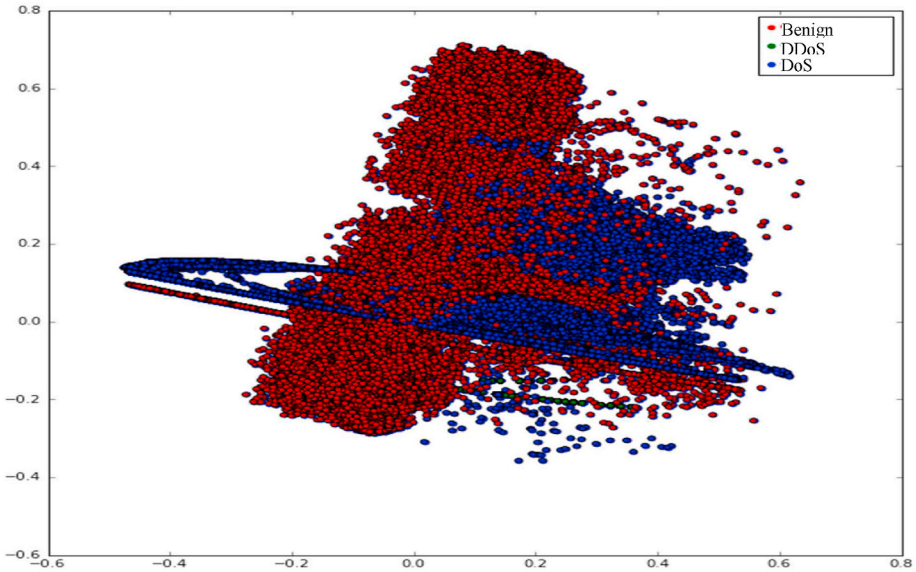


Fig. 6. Classes representation of the CSE-CIC-IDS2018 dataset.

Table 1
Comparative analysis of the methods on CSE-CIC-IDS2018 dataset.

	Homogeneity score	Completeness score	V-measure	Adjusted Rand Index	Adjusted Mutual Information	Silhouette scores	Estimated number of clusters
PCA + DBSCAN	0.8801	0.8797	0.8799	**0.8989**	0.8797	**0.5798**	3
PCA + Agglomerative	0.8994	0.9002	0.8998	**0.9130**	0.8993	**0.5405**	3
PCA + k-means	0.8864	0.8851	0.8858	**0.9094**	0.8851	**0.5833**	3
k-means	0.5696	0.5930	0.5811	0.5331	0.5696	0.4085	3
Agglomerative	0.5426	0.5577	0.5501	0.5171	0.5425	0.4772	3
DBSCAN	0.2482	0.1758	0.2058	0.0227	0.1756	0.4266	3

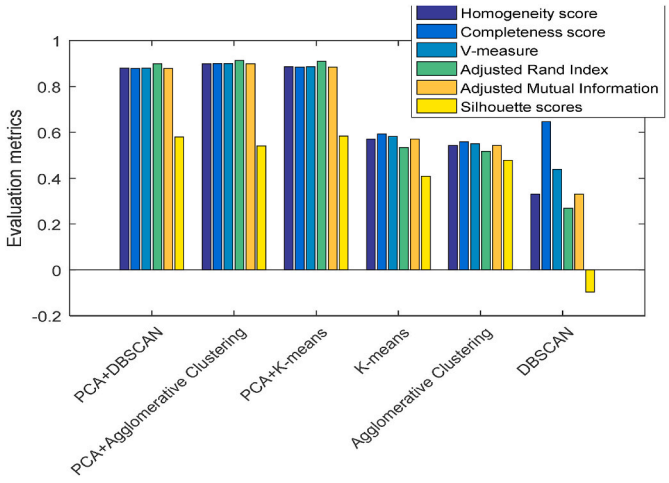


Fig. 7. Comparison of the methods on CSE-CIC-IDS2018 dataset.

of Azerbaijan National Academy of Sciences on the virtual machine with Ubuntu 16.04.3 LTS AMD64 system, 331.2-GB memory, and 2933.437 MHz CPU. The implementation of the method is conducted on Python and Tensorflow.

To provide the effective detection of DDoS attacks in the cloud environment, the classification model must be trained with large amounts of data. The CSE-CIC-IDS2018 dataset of the Canadian Institute of Cybersecurity is used to conduct the experiments [42,43]. This dataset is constructed on the user's behavior based on HTTPS, HTTP, SMTP, POP3, IMAP, SSH, and FTP protocols. The dataset is available on the Canadian Institute for Cybersecurity website.

In this dataset, the data is recorded separately in CSV files on days of the week. The data classes in each of these files are also different. In general, the dataset consists of one genuine traffic class and several most common network attack classes. DoS, DDoS, Brute Force, XSS, SQL Injection, Infiltration, Portscan, and Botnet are attack classes of the CSE-CIC-IDS2018 dataset. In this study, the CSV file consisting of three attack classes such as 0- Benign, 1- DDoS attack, and 2- DoS attack is taken. The data points of the three classes of this dataset are illustrated in Fig. 6. The number of rows of this file is 1048574.

The dataset is fully labeled and consists of 80 network traffic features.

In the application of the proposed approach to the CSE-CIC-IDS2018 dataset, the method achieved high values on various clustering metrics compared to existing methods (Table 1).

Here, the Adjusted Rand Index (ARI) determines the similarity degree of the samples included in the cluster. Therefore, the higher the ARI index, the better the clustering model is considered. In experiments, PCA + DBSCAN, PCA + Agglomerative, and PCA + k-means algorithms achieved high values on the ARI index and reached 0.8989, 0.9130, 0.9094 values, respectively. However, traditional k-means, Agglomerative, and DBSCAN algorithms achieved lower values on the ARI index and reached to 0.5331, 0.5171, and 0.0227 values, respectively. Note that ARI obtains values between 1 and -1. The approximation of the index to 0 means placing all points in random segments. Placing points in random segments are not considered the best case. The model with the high Silhouette value is considered a good clustering. The proposed approach also achieved high values on this metric.

For a better demonstration of the results in Table 1, Fig. 7 visually illustrates the comparison of methods.

As seen from Fig. 7, the proposed approach demonstrates better results compared to existing approaches.

In the process of proposed approach implementation, the algorithm has divided the CSE-CIC-IDS2018 dataset into three classes accurately. The visual illustration of the clustering results of each algorithm is represented in Fig. 8.

The purpose of the usage of the feature selection method in the approach is to provide the absence of features with zero elements in the dataset in the learning process and to reduce the size when using big

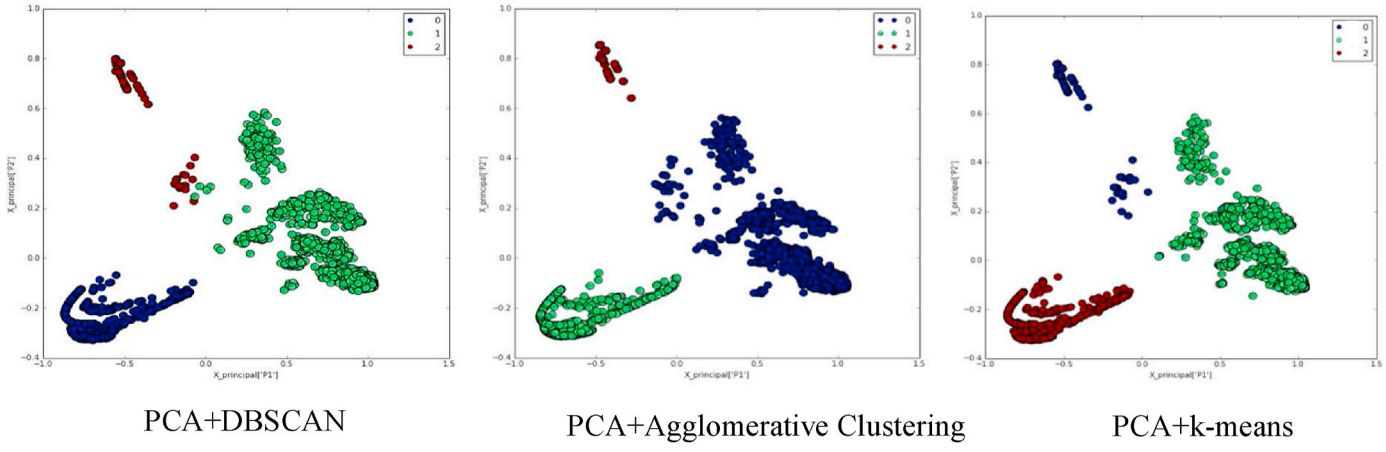


Fig. 8. Clustering results of the proposed approach.

Table 2
Selected features from the CSE-CIC-IDS2018 dataset.

#	Feature	Description	#	Feature	Description
1.	Dst Port	Destination Port	11.	Flow Pkts/s	flow packets rate that is number of packets transferred per second
2.	fl_dur	Flow Duration	12.	Flow IAT Mean	Average time between two flows
3.	Tot Fwd Pkts	Total packets in the forward direction	13.	Flow IAT Std	Standard deviation time two flows
4.	Tot Bwd Pkts	Total packets in the backward direction	14.	Flow IAT Max	Maximum time between two flows
5.	TotLen Fwd Pkts	Total size of packet in forward direction	15.	Flow IAT Min	Minimum time between two flows
6.	TotLen Bwd Pkts	Total size of packet in backward direction	16.	Fwd IAT Tot	Total time between two packets sent in the forward direction
7.	Fwd Pkt Len Max	Maximum size of packet in forward direction	17.	Fwd IAT Mean	Mean time between two packets sent in the forward direction
8.	Fwd Pkt Len Min	Minimum size of packet in forward direction	18.	Fwd IAT Std	Standard deviation time between two packets sent in the forward direction
9.	Fwd Pkt Len Mean	Average size of packet in forward direction	19.	Fwd IAT Max	Maximum time between two packets sent in the forward direction
10.	Flow Byts/s	Flow byte rate that is number of packets transferred per second	20.	Fwd IAT Min	Minimum time between two packets sent in the forward direction

Table 3
Accuracy, precision, recall, F1-score values of the methods on the CSE-CIC-IDS2018 dataset.

Methods	Class name	Accuracy	Precision	Recall	F1-score
PCA + DBSCAN	Benign (0)	0.9903	0.9532	0.9943	0.9721
	DDoS (1)	1.0000	0.9822	1.0000	0.9933
	DoS (2)	0.9421	1.0000	0.9432	0.9701
PCA + k-means	Benign (0)	0.9923	0.9441	0.9911	0.9741
	DDoS (1)	1.0000	0.9833	1.0000	0.9923
	DoS (2)	0.9412	1.0000	0.9432	0.9711
K-means	Benign (0)	0.3002	0.9503	0.3021	0.4541
	DDoS (1)	1.0000	0.2422	1.0000	0.3832
	DoS (2)	0.0301	0.0421	0.0334	0.0323
DBSCAN	Benign (0)	0.5723	0.0111	0.5732	0.0323
	DDoS (1)	0.0000	0.0000	0.0000	0.0000
	DoS (2)	0.4912	1.0000	0.4912	0.6643

data. Thus, about 20 features with non-zero elements from the CSE-CIC-IDS2018 dataset were used in the experiments. Note that the total number of features of this dataset is 80. The used features are listed in Table 2.

For the evaluation of the attack detection accuracy, the proposed method was evaluated on the basis of accuracy, precision, recall, and F1-score metrics, and their confusion matrix was constructed. The experiment results of the methods performed on the CSE-CIC-IDS2018 dataset are included in Table 3.

As seen from Table 3 on the CSE-CIC-IDS2018 dataset, the PCA + DBSCAN model has classified the data points of the Benign class with an accuracy of 0.9903, DDoS class with an accuracy of 1.0000, and DoS class with an accuracy of 0.9421. The algorithm has also demonstrated high results on other metrics. Here PCA + DBSCAN model has classified the data points of Benign, DDoS, and DoS classes with a precision score of 0.9532, 0.9822, and 1.0000, with a recall score of 0.9943, 1.0000, and 0.9432, and with an F1-score of 0.9721, 0.9933, and 0.9701, respectively. We obtain low results when applying a simple DBSCAN algorithm to the CSE-CIC-IDS2018 dataset. The algorithm could not recognize any of the points belonging to the DDoS class, and incorrectly assigned them to the Benign class by making a mistake in recognizing these points. The accuracy metric of the algorithm was low in the recognition of other class points. Here, the accuracy metric for Benign recognition was 0.5723 and for DoS recognition was 0.4912. Also, in the testing of the k-means, the algorithm was able to recognize points from the Benign and DoS classes with an accuracy of 0.3002 and 0.0301, respectively. In recognition theory, a model with an accuracy of less than 0.5 is considered bad. This algorithm also demonstrated low results on other metrics. When applying the PCA + k-means algorithm to the CSE-CIC-IDS2018 dataset, the results of the k-means algorithm were significantly improved. Thus, the PCA + k-means algorithm obtained the values of 0.9923, 1.0000, and 0.9412, respectively on the accuracy metric recognizing the points from the classes Benign (0), DDoS (1), and DoS (2) with high accuracy. This is a very good result in terms of recognition theory.

The confusion matrix obtained during the testing of algorithms on the CSE-CIC-IDS2018 dataset has been described in Fig. 9.

As seen from the confusion matrix of the DBSCAN algorithm described in Fig. 9, the algorithm made many mistakes in recognizing the samples. Thus, DBSCAN was able to identify samples from the 1-st class with an accuracy of 0.57, samples from the 2nd class with an accuracy of 0.0, and samples from the 3rd class with an accuracy of 0.49. Here, DBSCAN could not collect the points of the dataset fluently on the diagonal of the matrix. But PCA + DBSCAN was able to collect points on the diagonal of the matrix with high accuracy. This pattern is also observed in other algorithms.

Other datasets for testing the proposed method are the HTTP CSIC 2010 and NSL-KDD [44,45].

Representations of the both “HTTP CSIC 2010” and NSL-KDD

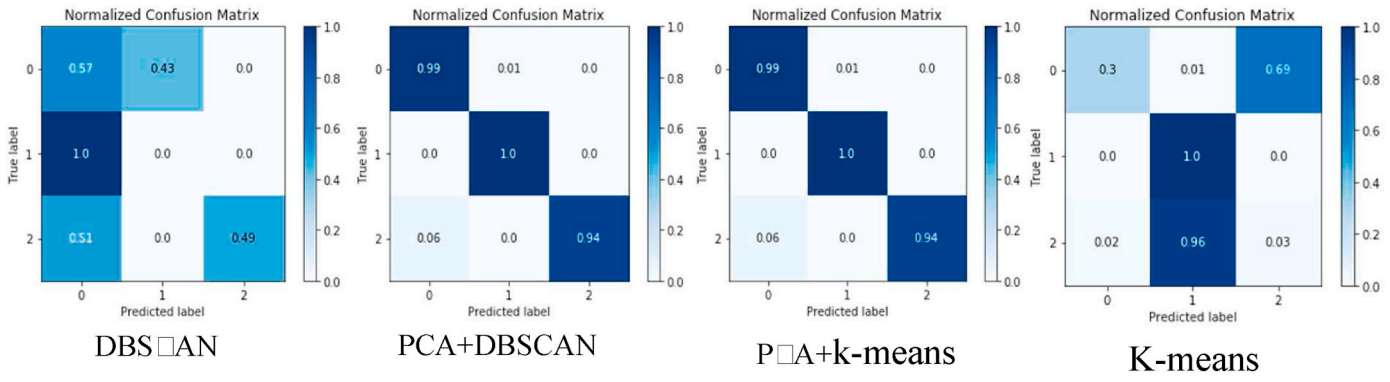


Fig. 9. Confusion matrix of the algorithms on the CSE-CIC-IDS2018 dataset.

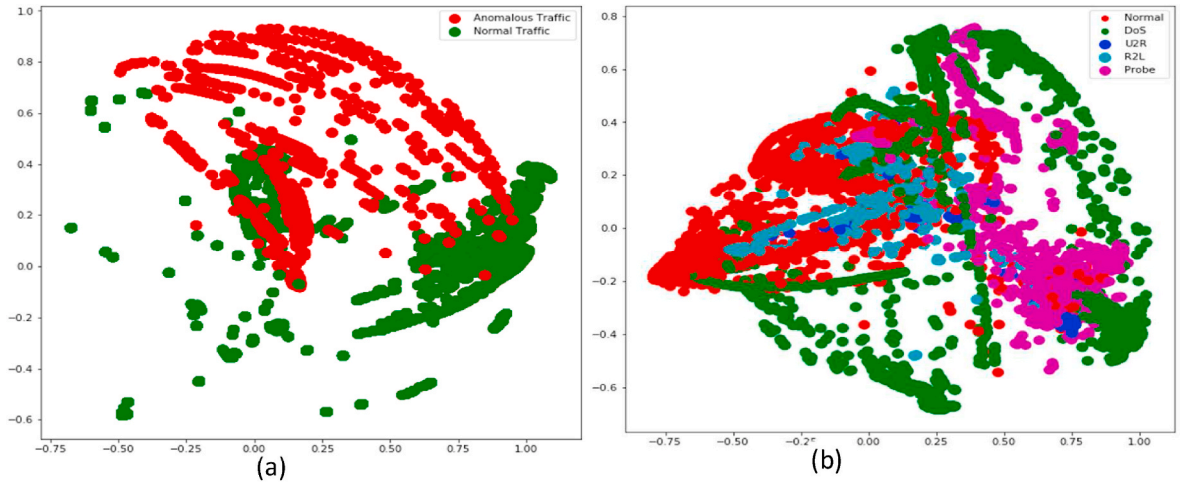


Fig. 10. Representations of the "HTTP CSIC 2010" (a) and NSL-KDD (b) datasets.

Table 4
Comparative analysis of the methods on the "HTTP CSIC 2010" dataset.

	Homogeneity score	Completeness score	V-measure	Adjusted Rand Index	Adjusted Mutual Information	Silhouette scores
PCA + DBSCAN	0.5997	1	0.7497	0.5834	0.5471	0.8245
PCA + Agglomerative	0.7322	0.7353	0.7337	0.7859	0.7321	0.3121
PCA + k-means	0.2845	0.9828	0.4413	0.7433	0.2836	0.3632
k-means	0.0849	0.0678	0.0754	0.1358	0.0678	0.3581
Agglomerative	0.2532	0.2572	0.2552	0.1306	0.2519	0.3548
DBSCAN	0.5545	0.9999	0.7133	0.4349	0.5025	0.7101

Table 5
Selected features from the "HTTP CSIC 2010" dataset.

#	Feature	Description	#	Feature	Description
1.	reqlen	Length of the request	9.	nletterpath	Number of letters char in the path *
2.	pathdepth	Length of the path	10.	ndigitpath	Number of digits in the path
3.	arglen	Length of the arguments	11.	notherpath	Number of other characters in path
4.	narg	Number of arguments	12.	typejs	Type js requested
5.	nletterarg	Number of letters in the arguments	13.	getflag	GET request flag
6.	ndigitarg	Number of digits in the arguments	14.	postflag	POST request flag
7.	notherarg	Number of other char in path	15.	putflag	PUT request flag
8.	nspecarg	Number of 'special' char in the arguments *			

datasets are illustrated in Fig. 10.

HTTP CSIC 2010 was developed at the "Information Security Institute" of CSIC (Spanish Research National Council). The dataset contains 36,000 normal requests, more than 25,000 anomalous requests, and 22 features. In the preprocessing stage of this research, the URLs in the dataset are encoded and they incorporate both normal and anomalous requests. For the application of the method into this dataset, the normalization of the dataset and feature selection are provided, and then the URL templates of the HTTP CSIC 2010 are fed into the input of the model. In experiments, a total of 71,484 samples are used. Table 4 represents the results of the provided experiments on the HTTP CSIC 2010 dataset.

In experiments on the "HTTP CSIC 2010" dataset, compared to the traditional algorithms proposed in this study the hybrid methods also achieved high results. Thus PCA + DBSCAN, PCA + Agglomerative, and PCA + k-means algorithms overall clustering metrics achieved high values. The higher Silhouette Coefficient score relates to a model with better defined clusters. Silhouette score of the proposed PCA + DBSCAN approach constituted into 0.8245, but in the traditional DBSCAN algorithm, this metric obtained a 0.7101 value.

About 15 features with non-zero elements were used from the HTTP CSIC 2010 dataset. The total number of features of this dataset is 21. The used features are listed in Table 5.

NSL-KDD is a Network based cyberattack dataset. This dataset contains 311,027 records and 41 types of features and they are assigned to attack or normal type. DoS (Denial of service), Probing (Surveillance and other probing attacks), U2R (Unauthorized access to the local superuser), and R2L (Unauthorized access from a remote machine) are four types of attack classes of the dataset. Table 6 represents the results of the experiments on the NSL-KDD dataset.

Table 6
Comparative analysis of the methods on the NSL-KDD dataset.

	Homogeneity score	Completeness score	V-measure	Adjusted Rand Index	Adjusted Mutual Information	Silhouette scores
PCA + DBSCAN	0.8020	0.4067	0.5397	0.4476	0.4061	0.3873
PCA + Agglomerative	0.4283	0.3588	0.3905	0.3719	0.3587	0.3672
PCA + k-means	0.3957	0.3335	0.3619	0.3653	0.3333	0.4435
k-means	0.0158	0.3311	0.0301	0.0121	0.0156	−0.2524
Agglomerative	0.3632	0.3096	0.3342	0.3403	0.3094	0.3752
DBSCAN	0.7617	0.3733	0.5011	0.3264	0.3724	0.2097

Table 7
Selected features from the NSL-KDD dataset.

#	Feature	Description	#	Feature	Description
1.	Count	No. of connections to the same host as the current connection in the last 2 s	11.	Root shell	1 if root shell is obtained; otherwise 0
2.	Destination bytes	Bytes sent from destination to source	12.	serror rate	Percentage of connections that have "SYN" errors
3.	diff srv rate	Percentage of connections to different services	13.	Service	Destination service (e.g. telnet, ftp)
4.	dst host count	Count of connections having the same destination hosts	14.	src bytes	Bytes sent from source to destination
5.	dst host diff srv rate	Source bytes	15.	srv count	No. of connections to the same service
6.	dst host rerror rate	Percentage of connections to the current host that have an RST error	16.	srv diff host rate	Percentage of connections to different hosts
7.	dst host srv rerror rate	Percentage of connections to the current host and specified service	17.	srv rerror rate	Percentage of connections that have "REJ" errors
8.	Num failed logins	No. of failed logins	18.	srv serror rate	Percentage of connections that have "SYN" errors
9.	Num file creations	No. of file creation operations	19.	Duration	Duration of the active connection
10.	rerror rate	Percentage of connections that have "REJ" errors	20.	Wrong fragment	No. of wrong fragments

As seen in Table 6, the proposed hybrid clustering algorithms have higher scores on all metrics than classic algorithms. The higher values of the Homogeneity score and Completeness score metrics of the clustering algorithms indicate better clustering. While the PCA + DBSCAN algorithm achieved the value of 0.8020 on the Homogeneity score metric, the value of the classical DBSCAN algorithm on this metric was 0.7617. The value of the PCA + DBSCAN algorithm on the Completeness score metric was still higher than the DBSCAN and they achieved 0.4067 and 0.3733 values, respectively. In general, among the clustering algorithms, the PCA + DBSCAN algorithm represents the highest results in the experimental study on all three datasets. The ability of the DBSCAN algorithm to adjust parameters (epsilon and minPts) further increases its ability to achieve high results in the accurate separation of the dataset. The methods of getting different results from multiple datasets depend on the structure of the data.

About 20 features with non-zero elements were used from the NSL-KDD dataset. The total number of features of this dataset is 41. The used features are given in Table 7.

Table 8 demonstrates the comparison of our results with those obtained by other authors on the various datasets.

The results we have obtained with PCA + DBSCAN and PCA + k-means methods are superior to methods proposed by other authors. In all models, the accuracy obtained a value above 0.7200. In our work, we obtained better results compared to the results of best-performing models in terms of accuracy, precision, recall, and F-measure metrics. In some works which are being compared, there is no information about what type of attacks are being detected. From the comparison, it is evident that the overall results show the appropriateness of applying PCA + DBSCAN and PCA + k-means methods to detect attacks in the datasets. The combination of these methods with the PCA, as depicted in Table 8, while increasing slightly the performance obtained and retains the results comparable to results from similar works.

8. Conclusion

Protection against DDoS attacks should not be only the responsibility of the cloud provider. The protection procedure should be provided comprehensively on both sides, both, by customers and the provider. Security must be comprehensive, from the device used by the user to access the network to the servers running in the cloud.

To protect the cloud environment from both internal and external attacks, based on data clustering, the method of detecting the DDoS attacks with high accuracy was proposed. The hybrid approach proposed in this study is based on the combination of machine learning methods. In this approach, the PCA extracts the most informative features among numerous features. By applying the clustering methods to a new dataset formed from the selected features, the DDoS attack detection was carried out.

The effectiveness of the proposed method was evaluated on the CSE-CIC-IDS2018, NSL-KDD, and HTTP CSIC 2010 datasets and a comparative analysis with the existing methods was performed. As the result, PCA + DBSCAN clustering performed well enough on all three datasets to separate data that has similar features with outliers and noises if the dataset features and algorithm parameters are selected well. The parameters of the DBSCAN algorithm get distinct values depending on the features of the individual datasets.

An architecture reflecting the application of the DDoS detection system in a cloud environment was constructed. In this architecture, the DDoS attack detection system is installed on each processing server of the cloud and it monitors the network traffic coming from the virtual network, internal network, and external network related to the appropriate virtual machine.

Funding details

This work was supported by the Science Development Foundation under the President of the Republic of Azerbaijan – Grant No. EIF-BGM-4-RFTF-1/2017–21/08/1.

Credit author statement

Fargana J. Abdullayeva: Idea generation, review, editing, read and

Table 8
Related work results.

Reference	Methods	Datasets	Accuracy	Precision	Recall	F-measure
[32]	SaE-ELM	NSL-KDD	0.8600	0.9500	–	0.8600
[33]	OCSA + RNN	NSL KDD	0.9400	0.9800	0.9500	0.9300
[36]	Deep Neural Network	CSE-CIC-IDS2018	0.7200	0.7700	0.8600	0.8100
[37]	Deep autoencoder	CSE-CIC-IDS2018	0.9700	–	–	–
	Restricted Boltzmann machine	CSE-CIC-IDS2018	0.9700	–	–	–
	Deep belief network	CSE-CIC-IDS2018	0.9700	–	–	–
Proposed model	PCA + DBSCAN	CSE-CIC-IDS2018	1.0000	0.9822	1.0000	0.9933
Proposed model	PCA + k-means	CSE-CIC-IDS2018	1.0000	0.9833	1.0000	0.9923

approve the final manuscript, data analysis, figures drawing, and tables design.

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.

References

[1] Osanaiye O, Choo KR, Dlodlo M. Distributed denial of service (DDoS) resilience in cloud: review and conceptual cloud DDoS mitigation framework. J Netw Comput Appl 2016;67:147–65.

[2] The treacherous 12–top threats to cloud computing + industry insights. Cloud Security Alliance (CSA); 2017. p. 59. https://downloads.cloudsecurityalliance.org/assets/research/top-threats/treacherous-12-top-threats.pdf.

[3] Golodoniuc P, Car NJ, Klump J. Distributed persistent identifiers system design. Data Sci J 2017;16(34):1–12. https://doi.org/10.5334/dsj-2017-034.

[4] Manavi MT. Defense mechanisms against distributed denial of service attacks: a survey. Comput Electr Eng 2018;72:26–38.

[5] 2019 Global DDoS threat landscape report, https://www.imperva.com/blog/2019-global-ddos-threat-landscape-report/.

[6] https://usa.kaspersky.com/about/press-releases/2020_kaspersky-research-finds-ddos-attacks-almost-double-in-q4-2019.

[7] Worldwide infrastructure security report (WISR). Mar 20th; 2019. https://www.netscout.com/blog/cloud-crosshairs.

[8] Khattak S, Ramay NR, Khan KR, Syed AA, Khayam SA. A taxonomy of botnet behavior, detection, and defense. IEEE Commun. Survey Tutorial 2014;16(2):898–924.

[9] DDoS attack that disrupted internet was largest of its kind in history, experts say, https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet.

[10] Zekri M, Kafhali SE, Aboutabit N, Saadi Y. DDoS attack detection using machine learning techniques in cloud computing environments. In: 3rd international conference of cloud computing technologies and applications. CloudTech; 2017. p. 1–8.

[11] Alyas M, Noor IM, Hassan H. DDOS attack detection strategies in cloud a comparative study. VFAST Transact. Soft. Eng. 2017;12(3):35–42. https://doi.org/10.21015/vtse.v12i3.502.

[12] Wang B, Zheng Y, Lou W, Hou YT. DDoS attack protection in the era of cloud computing and Software-Defined Networking. Comput Network 2015;81:308–19.

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

[14] Bojovic PD, Bašicevic I, Ocovaj S, Popovic M. A practical approach to detection of distributed denial-of-service attacks using a hybrid detection method. Comput Electr Eng 2019;73:84–96.

[15] Abdullayeva FJ. Detection of cyberattacks in cloud computing service delivery models using correlation based feature selection. IEEE 15th Int. Conf. Appl. Info. Commun. Technol. (AICT) 2021:1–4. https://doi.org/10.1109/AICT52784.2021.9620347.

[16] Aamir M, Zaidi SM. Clustering based semi-supervised machine learning for DDoS attack classification. J. King Saud Univ. Comp. Info. Sci. 2021;33(4):436–46. https://doi.org/10.1016/j.jksuci.2019.02.003.

[17] Rawashdeh A, Alkasassbeh M, Al-Hawawreh M. An anomaly-based approach for DDoS attack detection in cloud environment. Int J Comput Appl Technol 2018;57(4):312–24.

[18] Cha B, Kim J. Study of multistage anomaly detection for secured cloud computing resources in future Internet. In: Proc. Of the ninth IEEE international conference on dependable, autonomic and secure computing; 2011. p. 1047–51. https://doi.org/10.1109/DASC.2011.171.

[19] Doua W, Chena Q, Chen J. A confidence-based filtering method for DDoS attack defense in cloud environment. Future Generat Comput Syst 2013;29:1838–50.

[20] Somani G, Gaur MS, Sanghi D, Conti M. DDoS attacks in cloud computing: collateral damage to non-targets. Comput Network 2016;109(2):157–71.

[21] Mishra P, Pilli ES, Varadharajan V, Tupakul U. Intrusion detection techniques in cloud environment: a survey. J Netw Comput Appl 2017;77:18–47. https://doi.org/10.1016/j.jnca.2016.10.015.

[22] Patil R, Dudeja H, Modi C. Designing an efficient security framework for detecting intrusions in virtual network of Cloud Computing. Comput Secur 2019;85:402–22.

[23] Wang M, Lu Y, Qin J. A dynamic MLP-based DDoS attack detection method using feature selection and feedback. Comput Secur 2020;88:1–15.

[24] Chiba Z, Abghour N, Moussaid K, El omri A, Rida M. Intelligent approach to build a Deep Neural Network based IDS for cloud environment using combination of machine learning algorithms. Comput Secur 2019;86:291–317.

[25] Al-Mamory SO, Algelal ZM. A modified DBSCAN clustering algorithm for proactive detection of DDoS attacks. In: Annual conference on new trends in information & communications technology applications. NTICT; 2017. p. 304–9. https://doi.org/10.1109/NTICT.2017.7976107.

[26] Dincalp U, Güzel MS, Sevinc O, Bostanci E, Askerzade I. Anomaly based distributed denial of service attack detection and prevention with machine learning. In: 2nd IEEE international symposium on multidisciplinary studies and innovative technologies. ISMSIT; 2018. p. 1–4.

[27] Karim AM, Güzel MS, Tolun MR, Kaya H, Çelebi FV. A new generalized deep learning framework combining sparse auto-encoder and Taguchi method for novel data classification and processing. Math Probl Eng 2018;2018:1–13. https://doi.org/10.1155/2018/3145947.

[28] Luo X, Di X, Liu X, Qi H, Li J, Cong L, Yang H. Anomaly detection for application layer user browsing behavior based on attributes and features. J Phys Conf 2018;1069(1):1–10.

[29] Kushwah GS, Ranga V. Voting extreme learning machine based distributed denial of service attack detection in cloud computing. J Inf Secur Appl 2020;53:1–12.

[30] Bhardwaj A, Mangat V, Vig R. Effective mitigation against IoTs using super materials for distributed denial of service attacks in cloud computing 2020;28(3):1359–62.

[31] Aydın H, Orman Z, Aydın MA. A long short-term memory (LSTM)-based distributed denial of service (DDoS) detection and defense system design in public cloud network environment. Comput Secur 2022;118:1–14.

[32] Kushwah GS, Ranga V. Optimized extreme learning machine for detecting DDoS attacks in cloud computing. Comput Secur 2021;105:1–21.

[33] Theja RS, Shyam GK. An efficient metaheuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment. Appl Soft Comput 2021;100:1–11.

[34] Abdullayeva FJ. Convolutional neural network-based automatic diagnostic system for AL-DDoS attacks detection. Int J Cyber Warf Terror (IJCWT) 2022;12(1):1–12. 11, In press.

[35] Vissers T, Somasundaram TS, Pieters L, Govindarajan K, Hellinckx P. DDoS defense system for web services in a cloud environment. Future Generat Comput Syst 2014;37:37–45.

[36] Amaizu GC, Nwakanma CI, Lee JM, Kim DS. Investigating network intrusion detection datasets using machine learning. In: Proc. of the IEEE International Conference on Information and Communication Technology Convergence; 2020. p. 1325–8.

[37] Ferrag MA, Maglaras L, Moschoyiannis S, Janicke H. Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study. J Inf Secur Appl 2020;50:1–19.

[38] Abdullayeva FJ. Advanced persistent threat attack detection method in cloud computing based on autoencoder and softmax regression algorithm. Array 2021;10:1–11.

[39] Bertino E, Islam N. Botnets and Internet of Things security. Computer 2017;50(2):76–9.

[40] Osanaiye O. IP spoofing detection for preventing DDoS attack in Cloud Computing. In: Proc. Of the 18th IEEE international conference on intelligence in next generation networks. ICIN; 2015. p. 139–41.

[41] Khattak S, Ramay NR, Khan KR, Syed AA, Khayam SA. A taxonomy of botnet behavior, detection, and defense. IEEE communications surveys & tutorials 2014;16(2):898–924.

[42] Sharafaldin I, Lashkari AH, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proc. Of the 4th international conference on information systems security and privacy; 2018. p. 108–16.

[43] CSE-CIC-IDS2018 on AWS, https://www.unb.ca/cic/datasets/ids-2018.html.

[44] HTTP DATASET CSIC 2010, Information Security Institute, http://www.isi.csic.es/dataset/.

[45] NSL-KDD dataset, UNB, https://www.unb.ca/cic/datasets/nsl.html.