Cluster Computing
https://doi.org/10.1007/s10586-020-03120-3

A statistical class center based triangle area vector method for detection of denial of service attacks

N. G. Bhuvaneswari Amma[1] · S. Selvakumar[2]

Received: 21 March 2019 / Revised: 14 March 2020 / Accepted: 28 April 2020

Abstract
Denial of service (DoS) attack is the menace to private cloud computing environment that denies services provided by cloud servers leading to huge business losses. Efficient DoS attack detection mechanisms are demanded which necessitates the extraction of features for its best performance. The lacuna in the existing feature extraction based detection systems is the sensitiveness of initial cluster center which leads to high false alarm rate and low accuracy. In this paper, this issue is addressed by proposing a class center based triangle area vector (CCTAV) method which computes the mean of target classes individually and extracts the correlation between features. Mahalanobis distance measure is used for profile construction and DoS attacks detection. The proposed CCTAV method is tested with five publicly available datasets and compared with existing methods. It is noticed that the proposed statistical method reduces the complexity of feature extraction and enhances the attack detection process. The proposed approach is evaluated by conducting tenfold cross validation to compute 95% confidence interval. It is evident that the accuracy obtained for all the datasets are within the confidence interval. Further, the proposed CCTAV method provides significant results compared to the state-of-the-art attack detection methods.

Keywords Attack detection · Cluster center · Denial of service attacks · Feature extraction · Mahalanobis distance · Statistical method

1 Introduction

Nowadays cloud computing plays an essential part of organizations to survive their businesses in the technological world as it caters to low cost computing facilities and services on demand [1]. The cloud infrastructure depends on the service provider to facilitate the usage of computing resources for satisfying the demands of users. The usage of services on the Internet using shared resources are vulnerable to cyber-attacks [2]. DoS attack is one of the most prevailing cyber-attacks and is dangerous in private cloud environments which has only limited resources [3]. The easy availability of DoS attack tools tempts the attackers to launch attacks even without any reasons. These attacks are launched using zombie machines called botnets. The fast-emerging Internet of things (IoT) may act as a backdoor for distributed DoS (DDoS) attacks through millions and millions of devices. The DDoS toolkit developers build botnet army which comprises of huge number of botnet zombies that massively attack the victim machines [4, 5]. The attackers mainly target on the services provided by cloud and also network bandwidth, I/O bandwidth, CPU, memory, etc. Nowadays multivector DDoS attacks are prevailing, in which applications, network, and bandwidth are simultaneously targeted to deny access to legitimate users. These attacks damage the services running on the affected cloud servers. Therefore, an effective detection mechanism is needed to protect the services of private

✉ N. G. Bhuvaneswari Amma
ngbhuvaneswariamma@gmail.com

S. Selvakumar
ssk@nitt.edu; director@iiitu.ac.in

[1] Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, Tamil Nadu 620 015, India

[2] Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli and Indian Institute of Information Technology, Una, Himachal Pradesh 177 005, India

Published online: 06 May 2020

cloud environments. According to Akamai (State of the Internet)/Security Q3 2017 [6], there is an 8% increase of layer 3 and 4 DDoS attacks compared to Q2 2017.

Defense of DoS attacks in cloud environments, based on the approach to tackle anomalies, are categorized into attack prevention, detection, and mitigation [1]. DoS attack prevention is a pro-active measure in which the requests of suspected attackers are filtered before affecting the server. Detection of attacks happens during the presence of attack signs on the server and the decreased availability of its services. The attack signs may be initial signs in which the attack has just started, or already the attack deteriorated the performance of the server. DoS attack mitigation is a reactive measure to keep the server alive, which is under attack. The defense mechanism considered in this study is detection of DoS attacks to classify the incoming traffic as either normal or attack.

DoS attack detection mechanisms based on techniques are classified into misuse based detection and anomaly based detection [7]. Misuse based detection techniques detect known attacks using signatures of those attacks. But these techniques frequently require manual update of the database with rules and signatures and are unable to detect unknown attacks. The anomaly based detection techniques model the normal behavior of the system and identifies anomalies as deviations from normal behavior. Since these detection techniques are constructed using normal network traffic profiles, these can detect unknown attacks as well. Because of detecting previously unseen attacks, most of the researchers have been attracted to adopt anomaly based technique for attack detection [8].

In this study, anomaly based approach is proposed for detection of DoS attacks. The anomaly based detection mechanisms are classified into statistical and machine learning based approaches [9]. The statistical approaches compute the mean and standard deviation of normal traffic and use the distance measures to compute the normal traffic profile [10]. The advantages of anomaly based statistical approaches are less detection time and low computational cost. The machine learning approaches learn the traffic using classification algorithms [11] and ensemble classifiers [12]. Recently, machine learning approaches have been used for attack detection by injecting attack data with the normal data for evading the detection using binary classifiers [13, 14]. The advantages of anomaly based machine learning approaches are high accuracy and non-requirement of prior assumptions. But the training time is more to build a learned model and also the computational complexity is high. In order to detect the attack early, anomaly based statistical approaches are preferred as the detection time is less compared to machine learning approaches which motivated us to use statistical approach in this study.

Further, to enhance the accuracy of the attack detection system and to speed up the attack detection process, feature extraction techniques are proposed in the literature [8, 15]. These techniques provide linear or nonlinear combinations of the original features. The feature extraction mechanisms also find the correlation between features to enhance the performance of attack detection systems. It provides transformed features which is the combination of all original features. The feature extraction methods based on geometrical approaches include triangle area map (TAM) [15], cluster center and nearest neighbor (CANN) [16], and geometric area analysis (GAA) using trapezoidal method [17]. These methods compute the cluster centers for the data and provide discrimination capabilities for recognizing the attack patterns. The challenge exists in cluster center based feature extraction mechanisms is the choice of initial cluster centers. This motivated us to propose a feature extraction mechanism that computes the centers based on traffic classes.

The following are the key contributions of this study:

1. Class center based triangle area vector (CCTAV) method is proposed to compute the class centers of training data and to extract the correlation among features to improve the detection accuracy and to reduce the false alarm rate.
2. Algorithm to generate triangle area vector (TAV) of each record.
3. Algorithm to construct normal traffic profile which comprises of mean and standard deviation of Mahalanobis distance (MahD) of all normal traffic.
4. Algorithm to detect DoS attacks using MahD based statistical approach.

The rest of the paper is organized as follows: Sect. 2 discusses the related work with respect to DoS attacks, feature extraction mechanisms, and attack detection mechanisms. Section 3 describes the proposed statistical class center based triangle area vector method. Section 4 describes the performance evaluation of the proposed method. Section 5 concludes the paper with the directions for further research. Appendix describes the illustration of the proposed method.

2 Related work

2.1 DoS attacks

DoS attack is an assault to make the online services unavailable to legitimate users [18–20]. The DoS attack in private cloud servers is dangerous as it is managed internally by an organization with limited resources. The cost of the attack is high in private cloud computing environment

[21, 22]. This attack is classified into three categories, e.g., volume based attack, protocol based attack, and application layer attack. The volume based attacks flood the victim machines with useless packets and saturate the bandwidth of the attacked machine or site. The protocol based attack consumes server resources, firewalls, and load balancers. The application layer attacks crash the web servers. These attacks threaten the individual private cloud environment leading to financial losses and detection of these attacks is challenging [23].

From the literature, it is observed that there is a lack of public DoS attack datasets that reveal recent attack patterns. The reason is that the generation of attack records requires deep traffic inspection which reveals sensitive confidential communications. The publicly available DoS attack datasets include KDD Cup 99 [24], NSL KDD [25], UNSW NB15 [26], CICIDS 2017 [27], and CSE-CIC-IDS 2018 [28].

2.2 Feature extraction mechanisms

The detection accuracy of attack detection systems is enhanced using feature extraction mechanisms. Various feature extraction techniques are discussed in the literature such as TAM, distance sum based support vector machine (DS-SVM), CANN, and GAA based anomaly detection system (ADS). The TAM method is used to enhance the attack detection process [8, 15, 16, 29]. The disadvantage of TAM is that each triangle is constructed by taking two points from the final cluster centers of *k*-means clustering. In *k*-means clustering, the choice of finding and fixing the seed itself is a problem and various initialization methods are discussed in the literature [30]. The DS-SVM method is used to find the correlation between the data sample and cluster centers. This approach also uses *k*-means clustering [31]. The CANN approach used the cluster centers and nearest neighbors for extracting the features [16]. The GAA based ADS extracted the features using principal component analysis which fails to identify the recent attacks leading to high false alarm rate [17]. In [32], covariance feature space is computed for extracting the correlation between features used for intrusion detection. The clustering based feature extraction mechanisms lack to fix the initial cluster centers. The lacuna of fixing the initial cluster center for extracting the feature motivated us to propose a feature extraction method using class center (CC) approach.

2.3 Attack detection mechanisms

Attack detection mechanisms are classified into statistical [10] and machine learning [11] mechanisms. The statistical method based detection mechanisms require less detection time. These methods compute the mean and standard deviation of normal profiles and check whether the traffic deviates from normal behavior. These detection mechanisms are based on distance measures and do not require prior knowledge about attacks. The distance measures are categorized into power based distances, distribution law based distances, and correlation based distances [9]. The most widely used distance measure [33] are Hellinger distance (HD), Euclidean distance (ED), MahD, etc. HD is used to reduce the features without degradation in performance [34]. ED is used to extract the correlation between features [35] and to find the cluster centers [15]. MahD and earth movers distance (EMD) are used to classify the attack traffic and the normal traffic [8, 29]. From the literature, it is evident that the distance measures are used for feature selection, classification, and clustering.

Among all the above distance measures, MahD is capable of measuring the dissimilarity between traffic records. Unlike ED and MD, the MahD evaluates the distance between two multivariate data objects by considering the correlations between variables and removing the dependence among variables. In the proposed approach, MahD measure is used for profile construction and classification due to its significance of measuring the distance between a point and a distribution, i.e., record and mean of normal traffic records [8, 36].

3 Statistical class center based triangle area vector method

The proposed statistical CCTAV method extracts the correlation between features based on the centers of the target classes, i.e., normal and DoS attacks. The extracted features using CCTAV are given as input to statistical DoS attack detection method. The detection is achieved by using MahD measure. Figure 1 depicts the block schematic of the proposed statistical CCTAV method. The proposed method consists of training and testing phases. The training phase comprises of CC computation, TAV generation, and profile construction process and creates a normal traffic profile. The testing phase comprises of TAV generation of test data and attack detection which uses the previously generated normal traffic profile. Deviation if any is detected as attack.

3.1 Dataset representation

Let T_r be the training dataset with n records comprising of T_{r_p} normal traffic records and T_{r_q} attack traffic records. Each record consists of k features with raw data. The attributes of the traffic datasets include network flow and

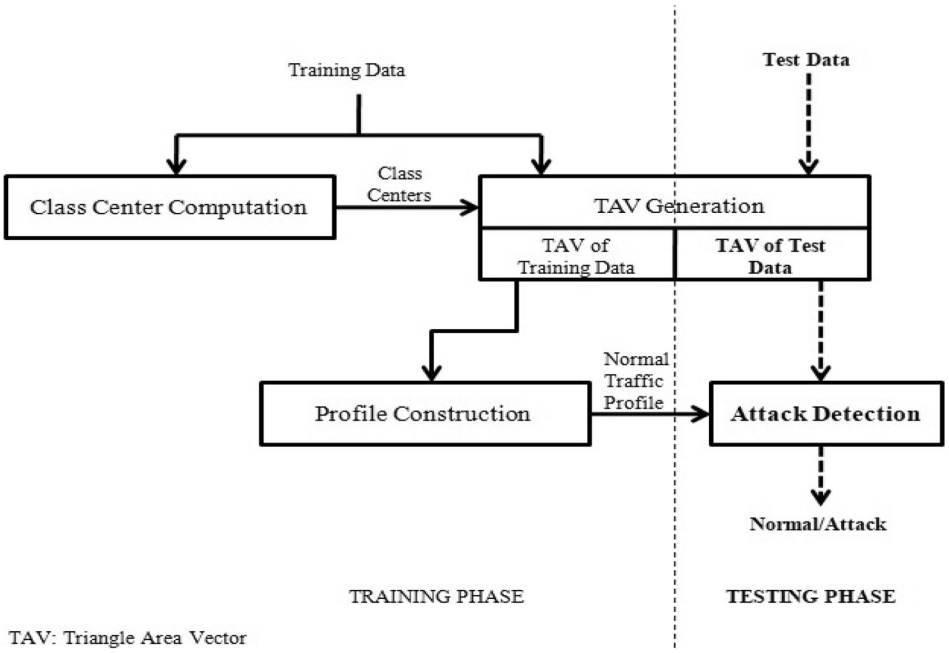

Fig. 1 Block schematic of proposed statistical CCTAV method

payload based features with continuous, binary, and nominal values. The nominal features are converted to numerical by replacing the values with ordered numbers [17] and the same has been used in this work. Moreover, the bias in the raw data is eliminated using min–max normalization [37]. The normalization process transforms the data into the range [0, 1] by finding the minimum and maximum value of each feature. The normalized value, NF_{ij} is computed as follows:

$$NF_{ij} = \frac{(RF_{ij} - min_j)}{(max_j - min_j)}\left(n_{max_j} - n_{min_j}\right) + n_{min_j} \tag{1}$$

where $RF_{ij}(1 \le i \le n, 1 \le j \le k)$ denotes the raw value of jth record and jth feature, min_j denotes the minimum value of the jth feature, max_j denotes the maximum value of the jth feature, n_{max_j} is the maximum value of the given new range, and n_{min_j} is the minimum value of the given new range.

3.2 Class center computation

The existing feature extraction methods compute the centers of classes using clustering approaches. These clustering approaches fail to find the correct cluster centers as the methodology involved in the choice of cluster seeds for clustering [30] is difficult. This increases the computational complexity of the feature extraction process. In order to reduce the computational cost involved in clustering to derive the centers of the target classes, a CC methodology is proposed to find the centers of the target classes.

The CC of each target class, C_c is computed by finding the mean of the records, R_{ic} of the corresponding class and is as follows:

$$C_c = \frac{\sum_{i=1}^{m} R_{ic}}{m} \tag{2}$$

where c ranges from 1 to n and n denotes the number of classes in the dataset.

3.3 Triangle area vector generation

The TAV method of feature extraction enhances the attack detection process. This method is based on triangle area based nearest neighbor approach [15]. As this method is based on the CCs, the deviation of anomaly is easy to find. As the proposed attack detection method is based on statistical approach, the TAVs of normal traffic records are only computed. This process generates $y = n(n-1)/2$ triangles for each record to form a TAV. Suppose the dataset consists of five classes then the record, R_i generates ten triangles, Tri_{10} as follows: $[R_iC_1C_2, R_iC_1C_3, R_iC_1C_4, R_iC_1C_5, R_iC_2C_3, R_iC_2C_4, R_iC_2C_5, R_iC_3C_4, R_iC_3C_5, R_iC_4C_5]$.

The perimeter of each triangle is computed by computing the ED between two points of the triangle. Consider the triangle, $R_iC_1C_2$ and the three sides of the triangle are R_iC_1, R_iC_2, and C_1C_2. The ED between two points are computed as follows:

$$ED(P_1, P_2) = \sqrt{\sum_{i=1}^{k} (P_{1_i} - P_{2_i})^2} \tag{3}$$

where k is the number of features, P_1 and P_2 are the points.

The three sides of a triangle, $R_iC_jC_k$ are computed as $S_1 = ED(R_i, C_j)$, $S_2 = ED(R_i, C_k)$, and $S_3 = ED(C_j, C_k)$. The perimeter of each triangle, P_{tri} is computed by the sum of the sides of the triangle and is shown as follows:

$$P_{tri} = S_1 + S_2 + S_3 \tag{4}$$

The sides of the triangle, S_1, S_2, and S_3 are computed by calculating the ED between two points that form the side of the triangle. The semi-perimeter, S_{peri} of the triangle is computed as follows:

$$S_{peri} = \frac{(S_1 + S_2 + S_3)}{2} \tag{5}$$

The triangle area, A_t is calculated as follows:

$$A_t = \sqrt{S_{peri}(S_{peri} - S_1)(S_{peri} - S_2)(S_{peri} - S_3)} \tag{6}$$

Therefore, the TAVs of training records of normal traffic are generated and represented as follows:

$$[TAV_i] = [A_{t_{1i}} A_{t_{2i}} A_{t_{3i}} \ldots A_{t_{1oi}}] \tag{7}$$

Algorithm 1 depicts the TAV generation process. This algorithm takes the class centers and normal traffic records of training data as inputs and outputs TAV of normal traffic records. For a given normal traffic record, the triangles are formed using the class centers and the traffic record. The area of each triangle is formed by computing the perimeter and semi-perimeter of the triangle. The TAV is generated using the triangle areas generated for the normal traffic record. The generated TAV of all the normal traffic records are used for normal traffic profile construction process.

3.4 Profile construction

The profile generation procedure as discussed in [8] has been utilized for profile construction process of the proposed method. The normal traffic profile is constructed by computing the MahD between each TAV of normal training records and the mean of TAV of the normal training records. The mean of the TAVs is computed as follows:

$$\bar{\text{TAV}}_N = \frac{1}{l}\sum_{i=1}^{l} TAV_i \tag{8}$$

The MahD requires the computation of covariance matrix. The covariance matrix from the TAVs is computed by extracting the correlation between features and is shown as follows:

$$Cov_{mat} = \begin{pmatrix} Sig(A_{t_{1i}}, A_{t_{1i}}) & \ldots & Sig(A_{t_{1i}}, A_{t_{10i}}) \\ \vdots & \vdots & \vdots \\ Sig(A_{t_{1i}}, A_{t_{10i}}) & \ldots & Sig(A_{t_{10i}}, A_{t_{10i}}) \end{pmatrix} \tag{9}$$

The MahD between TAV of the record with the mean of the TAVs is computed as follows:

$$MahD_i\left(TAV_i, \bar{\text{TAV}}_N\right) = \sqrt{\left(TAV_i, \bar{\text{TAV}}_N\right)^T Cov_{mat}^{-1}\left(TAV_i, \bar{\text{TAV}}_N\right)} \tag{10}$$

The threshold is computed using mean and standard deviation of the MahD of all normal traffic and is computed as follows:

$$\bar{\text{MahD}}_N = \frac{1}{l}\sum_{i=1}^{l} MahD_i \tag{11}$$

Algorithm 1: TAV Generation Algorithm

Input: Class centers, C_c and normal traffic records, T_{r_p}

Output: Triangle area vector, TAV_i

Process:

1: **for** each normal traffic training record, $T_{r_{pi}}$ **do**
2: Generate triangles, Tri_q for each record
3: **for** each triangle **do**
4: Compute sides of triangle, (S_1, S_2, and S_3) using ED as in (3)
5: Compute perimeter of triangle, P_{tri} using (4)
6: Compute semi-perimeter of triangle, S_{peri} using (5)
7: Compute triangle area, A_t using (6)
8: **end for**
9: Formulate TAV using (7)
10: **end for**
11: **return** TAV_i

The standard deviation defines the spread of TAVs and is calculated as follows:

$$Sig_{MahD_N} = \sqrt{\frac{1}{l-1}\sum_{i=1}^{l}\left(MahD_i - \overline{MahD_N}\right)^2} \quad (12)$$

The profile for attack detection, Pr_{NT} is constructed using MahD mean and standard deviation of all normal traffic TAV. It is formulated as follows:

$$Pr_{NT} = \left(\overline{MahD_N}, Sig_{MahD_N}\right) \quad (13)$$

The steps to construct normal traffic profile is depicted in Algorithm 2. This algorithm takes the TAVs of normal traffic training data as input and outputs traffic profile. The mean of the TAVs and covariance matrix of each TAV are computed by extracting the correlation between features. Also, the MahD between TAV of normal traffic record with the TAV mean is computed to identify the dissimilarity between records. Then the traffic profile is constructed by computing mean and standard deviation of the MahD between TAV and mean of TAVs.

where α is the normal distribution parameter ranges from 1 to 3 with an increment of 1 [8]. It denotes the range that the traffic records are classified as normal in the normal distribution of the MahD trained during the profile construction process. The attack detection is made with certain level of confidence varying from 68 to 99.7% with the selection of different values of α. It is to be noted that there is no significance for α value below 1 and above 3.

3.6 Attack detection

The test traffic dataset consists of both normal and attack traffic records. The DoS attack is detected by computing the TAV of test data and MahD between TAV of test data and the mean of TAV_N which is computed in profile construction process. The process of attack detection is depicted in Algorithm 3. This algorithm takes the traffic profile, test traffic, and normal distribution parameter as inputs and detects whether the test traffic is normal or attack. The TAV of all the test traffic records is computed and if the MahD between TAV of test traffic record and the mean of TAVs ($\bar{}_{\mathrm{TAV}_N}$) lies within the threshold then the test traffic, tx_i is detected as normal traffic else detected as attack traffic.

Algorithm 2: Profile Construction Algorithm
Input: TAVs of normal traffic, TAV_i
Output: Traffic profile, Pr_{NT}

Process:
1: Compute the mean of TAVs, $\overline{TAV_N}$ using (8)
2: Generate covariance matrix, Cov_{mat} using (9)
3: **for** each TAV_i **do**
4: Compute MahD between TAV and $\overline{TAV_N}$ using (10)
5: **end for**
6: Compute mean of MahD of all normal traffic, $\overline{MahD_N}$ using (11)
7: Compute standard deviation, Sig_{MahD_N} using (12)
8: Formulate traffic profile, $Pr_{NT} = \left(\overline{MahD_N}, Sig_{MahD_N}\right)$
9: **return** Pr_{NT}

3.5 Threshold computation

The threshold to distinguish normal and attack traffic is computed as follows:

$$Thresh = \overline{MahD_N} \pm Sig_{MahD_N} \times \alpha \quad (14)$$

Algorithm 3: Attack Detection Algorithm
Input: Traffic profile, test traffic, and normal distribution parameter
Output: $Normal/Attack$

Process:
1: **for** each test traffic record, tx_i **do**
2: Generate TAV of test traffic
3: Compute MahD between TAV of test traffic and $\overline{TAV_N}$
4: Compute threshold, $Thresh$ using (14)
5: **if** the observed MahD lies within $Thresh$ **then**
6: The tx_i is $Normal$
7: **else**
8: The tx_i is $Attack$
9: **end if**
10: end for
11: **return** $Normal/Attack$

4 Experimental results and discussions

The proposed method was implemented in MATLAB R2018a, on IntelCore2 Quad CPU Q9650@3.00 GHz processor and 16 GB RAM under Windows 10. Experiments were performed on the available benchmark datasets such as KDD Cup, NSL KDD, UNSW NB, CICIDS, and CSE-CIC-IDS. The statistics of the datasets comprising of training and testing samples are tabulated in Table 1. These datasets consist of normal traffic and various attack traffic records. In this study, normal and DoS attack records are only considered for experimentation.

4.1 Description of benchmark datasets

This subsection describes the benchmark intrusion detection datasets used for evaluating the performance efficacy of the proposed method.

4.1.1 KDD Cup

The KDD Cup 99 dataset was generated by MIT Lincoln Labs by simulating U.S. Air Force environment in local-area network. They acquired nine weeks of raw TCP dump data with multiple attacks peppered in the environment. Seven weeks of network traffic was considered as raw training data with the size of about four gigabytes of compressed binary TCP dump data. This was generated as five million connection records. Similarly, 2 weeks of test data yielded around two million connection records. Each connection record consists of about 100 bytes. It comprises of 41 features in each record. The DoS attack traffic includes back, land, neptune, ping of death, smurf, teardrop, process table, and mail bomb. Data is partitioned into training and testing set having 488,736 and 289,650 records, respectively. Training set contains 97,278 normal and 391,458 DoS attack records while 60,593 normal records and 229,057 DoS attack records are found in testing set [24].

Table 1 Statistics of datasets

Dataset (year)	Training samples		Testing samples	
	Normal	DoS attack	Normal	DoS attack
KDD Cup (1999)	97,278	391,458	60,593	229,057
NSL KDD (2009)	13,449	9195	2152	3603
UNSW NB (2015)	20,520	4076	56,000	12,264
CICIDS (2017)	128,737	29,285	55,173	12,550
CSE-CIC-IDS (2018)	68,403	89,619	29,315	38,408

4.1.2 NSL KDD

The NSL KDD dataset is a refined version of KDD Cup dataset which is widely used as a standard dataset for attack detection. It consists of same number and similar type of DoS attacks as that of KDD Cup. Data is partitioned into training and testing set having 22,644 and 5755 records, respectively. Training set contains 13,449 normal and 9195 DoS attack records while 2152 normal records and 3603 DoS attack records are found in testing set [25].

4.1.3 UNSW NB

The UNSW NB dataset was generated by University of South Wales (UNSW) cyber security lab using a synthetic environment. They used IXIA tool to simulate the real normal and the synthetic abnormal network traffic in the synthetic environment. This dataset represents nine categories of attacks including DoS attacks. There are 48 features that have been extracted using Argus Bro-IDS tools. Data is partitioned into training and testing set having 24,596 and 68,264 records, respectively. Training set

contains 20,520 normal and 4076 DoS attack records while 56,000 normal records and 12,264 DoS attack records are found in testing set [26].

4.1.4 CICIDS

The CICIDS dataset was generated by Canadian Institute for Cybersecurity (CIC) consisting of normal and the most recent common attacks resembling the real-world data in pcap file format. It also includes the results of the network traffic analysis using CICFlowMeter with labeled flows. This dataset consists of 83 features consisting of 2,359,087 normal records and 294,506 DoS attack records. Among these 70% of the records are considered for training and 30% of the records are considered for testing. Training set contains 128,737 normal and 55,173 DoS attack records while 29,285 normal records and 12,550 DoS attack records are found in testing set [27, 38].

4.1.5 CSE-CIC-IDS

The CSE-CIC-IDS dataset was generated in a systematic manner by collaborative project between the communications security establishment (CSE) and the CIC, which contains detailed descriptions of intrusions and abstract distribution models for applications, protocols, or lower level network entities. This dataset consists of 78 features consisting of 97,718 normal records and 128,027 DoS attacks records. Among these 70% of the records are considered for training and 30% of the records are considered for testing. Training set contains 128,737 normal and 55,173 DoS attack records while 29,285 normal records and 12,550 DoS attack records are found in testing set [28, 38].

4.2 Performance evaluation

The performance evaluation of the DoS attack detection system is based on true negative (TN), false positive (FP), false negative (FN), and true positive (TP) measures. TN is the number of normal traffic records that are correctly classified as normal, FP is the number of normal traffic records that are incorrectly classified as attack, FN is the number of attack records that are incorrectly detected as normal, and TP is the number of attack records that are correctly detected as attack. The performance measures of the test data for DoS attack detection with varying thresholds for statistical attack detection with no feature extraction (FE), statistical attack detection with clustering based TAV (CTAV) method, and statistical attack detection with proposed CCTAV method are tabulated in Table 2. The reason for choosing these methods for comparison is that the proposed approach enhances the attack detection process by overcoming the limitation of clustering based feature extraction mechanism. The statistical attack detection with no FE uses the attack detection module of the proposed approach. The normal profile is constructed by computing the mean of normalized normal traffic data. The CTAV is similar to that of the proposed approach except that the centers for TAV generation are computed using *k*-means clustering.

The performance metrics, such as precision, recall/true positive rate (TPR), F-Measure, false positive rate (FPR), accuracy, and error rate (ER) are computed using (15), (16), (17), (18), (19), and (20) respectively and tabulated in Table 3. Precision is the ratio of correctly classified attack traffic to all the classified traffic.

$$\text{Precision} = \frac{TP}{TP + FP} \times 100 \tag{15}$$

Recall/TPR is the measure of the fraction of the attack class that was correctly detected.

$$\text{Recall/TPR} = \frac{TP}{TP + FN} \times 100 \tag{16}$$

F-Measure is the harmonic combination of the precision and recall into a single measure.

$$\text{F} - \text{Measure} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{17}$$

FPR measures the fraction of the normal class that was incorrectly classified as attack.

$$\text{FPR} = \frac{FP}{FP + TN} \times 100 \tag{18}$$

Accuracy measures the ability of the attack detection system that correctly classifies the class label of the test data.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \times 100 \tag{19}$$

ER measures the attack detection system that incorrectly classifies the class label of the test data.

$$\text{ER} = \frac{FN + FP}{TP + FP + TN + FN} \times 100 \tag{20}$$

The performance evaluation metrics are measured and plotted for varying thresholds, i.e., different values of normal distribution parameter for all the five datasets.

4.3 Feature extraction methods versus performance metrics

Figure 2 depicts the FE methods versus precision for the five datasets, such as KDD Cup, NSL KDD, UNSW NB, CICIDS, and CSE-CIC-IDS with varying thresholds. The

Table 2 Performance measures of test data

Dataset	Threshold	FE	TN	FP	FN	TP
KDD Cup	$\alpha = 1$	No FE	59,530	1063	10,967	218,090
		CTAV	59,774	819	9639	219,418
		CCTAV	59,915	678	7912	221,145
	$\alpha = 2$	No FE	59,620	973	10,004	219,053
		CTAV	59,866	727	8926	220,131
		CCTAV	60,111	482	6905	222,152
	$\alpha = 3$	No FE	59,888	705	8937	220,120
		CTAV	60,004	589	7924	221,133
		CCTAV	60,429	164	5871	223,186
NSL KDD	$\alpha = 1$	No FE	2026	126	82	3521
		CTAV	2069	83	60	3543
		CCTAV	2101	51	53	3550
	$\alpha = 2$	No FE	2058	94	69	3534
		CTAV	2093	59	56	3547
		CCTAV	2119	33	49	3554
	$\alpha = 3$	No FE	2083	69	57	3546
		CTAV	2101	51	52	3551
		CCTAV	2136	16	43	3560
UNSW NB	$\alpha = 1$	No FE	54,581	1419	8646	3618
		CTAV	54,812	1188	7191	5073
		CCTAV	55,071	929	6086	6178
	$\alpha = 2$	No FE	54,696	1304	8269	3995
		CTAV	54,974	1026	6933	5331
		CCTAV	55,129	871	5417	6847
	$\alpha = 3$	No FE	54,805	1195	8036	4228
		CTAV	55,027	973	6741	5523
		CCTAV	55,221	779	4638	7626
CICIDS	$\alpha = 1$	No FE	53,304	1869	1016	11,534
		CTAV	53,449	1724	852	11,698
		CCTAV	53,591	1582	739	11,811
	$\alpha = 2$	No FE	53,577	1596	836	11,714
		CTAV	53,798	1375	671	11,879
		CCTAV	53,954	1219	595	11,955
	$\alpha = 3$	No FE	53,979	1194	638	11,912
		CTAV	54,115	1058	529	12,021
		CCTAV	54,242	931	472	12,078
CIC-CSE-IDS	$\alpha = 1$	No FE	28,068	1247	942	37,466
		CTAV	28,129	1186	816	37,592
		CCTAV	28,352	963	675	37,733
	$\alpha = 2$	No FE	28,246	1069	758	37,650
		CTAV	28,321	994	681	37,727
		CCTAV	28,488	827	598	37,810
	$\alpha = 3$	No FE	28,369	946	837	37,571
		CTAV	28,533	782	706	37,702
		CCTAV	28,656	659	531	37,877

objective of FE methods is to enhance the attack detection process. It is evident from Fig. 2a–e, that higher precision is obtained for attack detection system with extracted features. It is observed that the proposed method detects anomaly better compared to the statistical attack detection without FE and CTAV method.

Table 3 Performance metrics of test data

Dataset	Threshold	FE	Precision	Recall	F-Measure	FPR	Accuracy	ER
KDD Cup	$\alpha = 1$	No FE	99.515	95.212	97.316	1.754	95.847	4.153
		CTAV	99.628	95.792	97.672	1.352	96.389	3.611
		CCTAV	99.546	96.546	98.095	1.119	97.034	2.966
	$\alpha = 2$	No FE	99.558	95.633	97.556	1.606	96.210	3.790
		CTAV	99.671	96.103	97.854	1.200	96.667	3.333
		CCTAV	99.784	96.985	98.365	0.795	97.450	2.550
	$\alpha = 3$	No FE	99.681	96.098	97.857	1.164	96.671	3.329
		CTAV	99.734	96.541	98.111	0.972	97.061	2.939
		CCTAV	99.927	97.437	98.666	0.271	97.916	2.084
NSL KDD	$\alpha = 1$	No FE	96.545	97.724	97.131	5.855	96.386	3.614
		CTAV	97.711	98.335	98.022	3.857	97.515	2.485
		CCTAV	98.584	98.529	98.556	2.370	98.193	1.807
	$\alpha = 2$	No FE	97.409	98.085	97.746	4.368	97.168	2.832
		CTAV	98.364	98.446	98.405	2.742	98.002	1.998
		CCTAV	99.080	98.640	98.860	1.533	98.575	1.425
	$\alpha = 3$	No FE	98.091	98.418	98.254	3.206	87.811	2.189
		CTAV	98.584	98.557	99.570	2.370	98.210	1.790
		CCTAV	99.553	98.807	99.178	0.743	98.975	1.025
UNSW NB	$\alpha = 1$	No FE	71.828	29.501	41.824	2.534	85.256	14.744
		CTAV	81.025	41.365	54.769	2.121	87.726	12.274
		CCTAV	86.928	50.375	63.786	1.659	89.724	10.276
	$\alpha = 2$	No FE	75.392	32.575	45.493	2.329	85.977	14.023
		CTAV	83.860	43.469	57.258	1.832	88.341	11.659
		CCTAV	88.715	55.830	68.532	1.555	90.789	9.211
	$\alpha = 3$	No FE	77.964	34.475	47.809	2.134	86.477	13.523
		CTAV	85.022	45.034	58.881	1.738	88.700	11.300
		CCTAV	90.732	62.182	73.792	1.391	92.065	7.935
CICIDS	$\alpha = 1$	No FE	86.055	91.904	88.884	3.388	95.740	4.260
		CTAV	87.155	93.211	90.082	3.125	96.196	3.804
		CCTAV	88.188	94.112	91.053	2.867	96.573	3.427
	$\alpha = 2$	No FE	88.009	93.339	90.596	2.893	96.409	3.591
		CTAV	89.626	94.653	92.071	2.492	96.979	3.021
		CCTAV	90.747	95.259	92.948	2.209	97.312	2.679
	$\alpha = 3$	No FE	90.890	94.916	92.859	2.164	97.295	2.705
		CTAV	91.911	95.785	93.808	1.918	97.657	2.343
		CCTAV	92.843	96.239	94.511	1.687	97.928	2.072
CIC-CSE-IDS	$\alpha = 1$	No FE	96.779	97.547	97.162	4.254	96.768	3.232
		CTAV	96.942	97.875	97.406	4.046	97.044	2.956
		CCTAV	97.511	98.243	97.876	3.285	97.581	2.419
	$\alpha = 2$	No FE	97.239	98.026	97.631	3.647	97.302	2.698
		CTAV	97.433	98.227	97.828	3.391	97.527	2.473
		CCTAV	97.860	98.443	98.150	2.821	97.896	2.104
	$\alpha = 3$	No FE	97.544	97.821	97.682	3.227	97.367	2.633
		CTAV	97.968	98.162	98.065	2.668	97.803	2.197
		CCTAV	98.290	98.617	98.453	2.248	98.243	1.757

Figure 3 depicts the FE methods versus recall for the five datasets, such as KDD Cup, NSL KDD, UNSW NB, CICIDS, and CSE-CIC-IDS with varying thresholds. It is evident from Fig. 3a–e that the obtained recall is high for attack detection system with extracted features. It is observed that the proposed method detects anomaly better

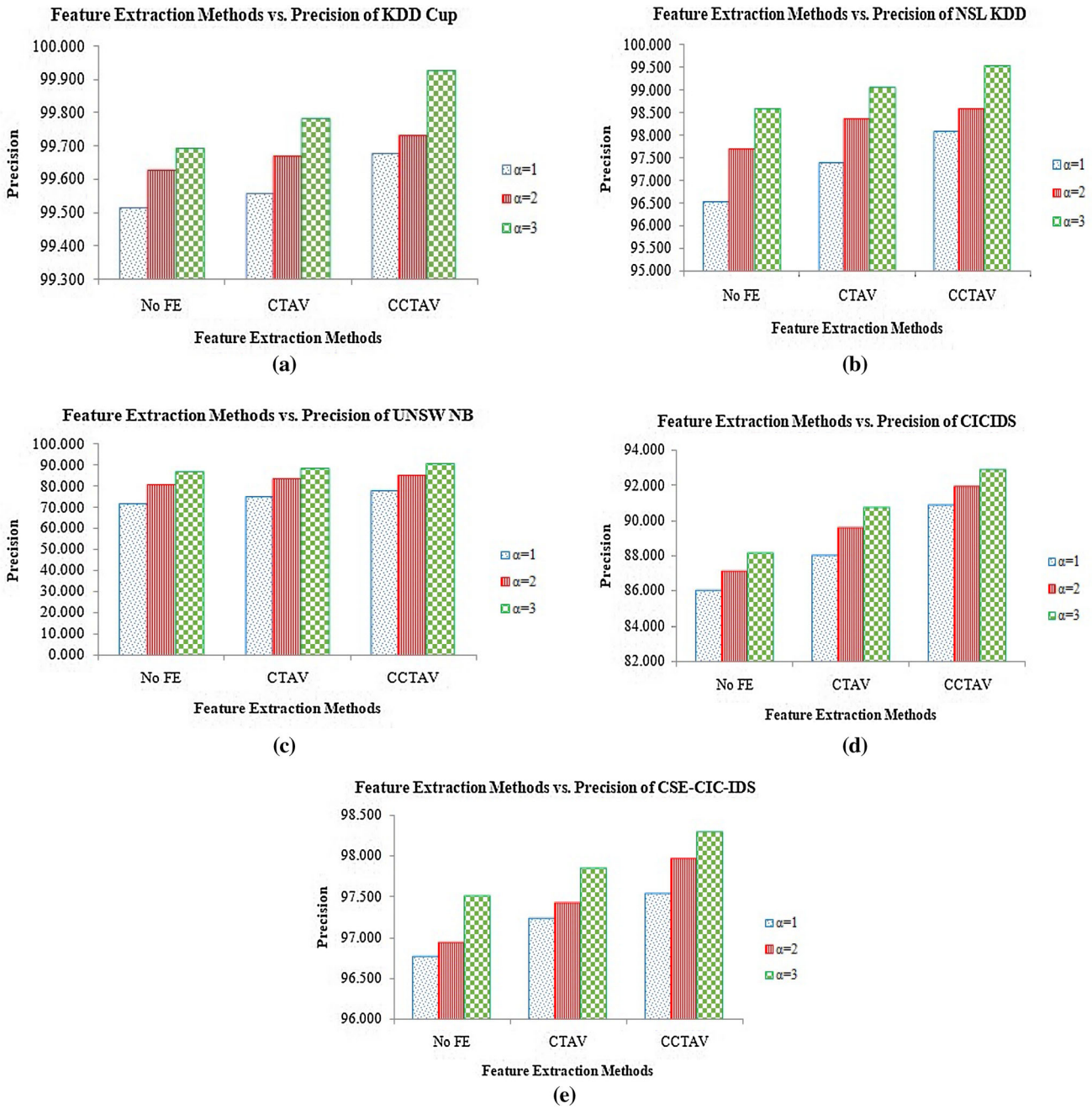

Fig. 2 Feature Extraction Methods vs. Precision. **a** KDD Cup, **b** NSL KDD, **c** UNSW NB, **d** CICIDS, **e** CSE-CIC-IDS

with high recall compared to the statistical attack detection without FE and CTAV method.

Figure 4 depicts the FE methods versus F-Measure for the five datasets, such as KDD Cup, NSL KDD, UNSW NB, CICIDS, and CSE-CIC-IDS with varying thresholds. It is evident from Fig. 4a–e that higher F-Measure is obtained for attack detection system with extracted features. It is observed that the proposed method detects anomaly better compared to the statistical attack detection without FE and CTAV method.

Figure 5 depicts the FE methods versus FPR for the five datasets, such as KDD Cup, NSL KDD, UNSW NB, CICIDS, and CSE-CIC-IDS with varying thresholds. It is evident from Fig. 5a–e that the obtained FPR is low for attack detection system with extracted features. It is observed that the FPR is low for the proposed method as it detects anomaly better compared to the statistical attack detection without FE and CTAV method.

Figure 6 depicts the FE methods versus accuracy for the five datasets, such as KDD Cup, NSL KDD, UNSW NB,

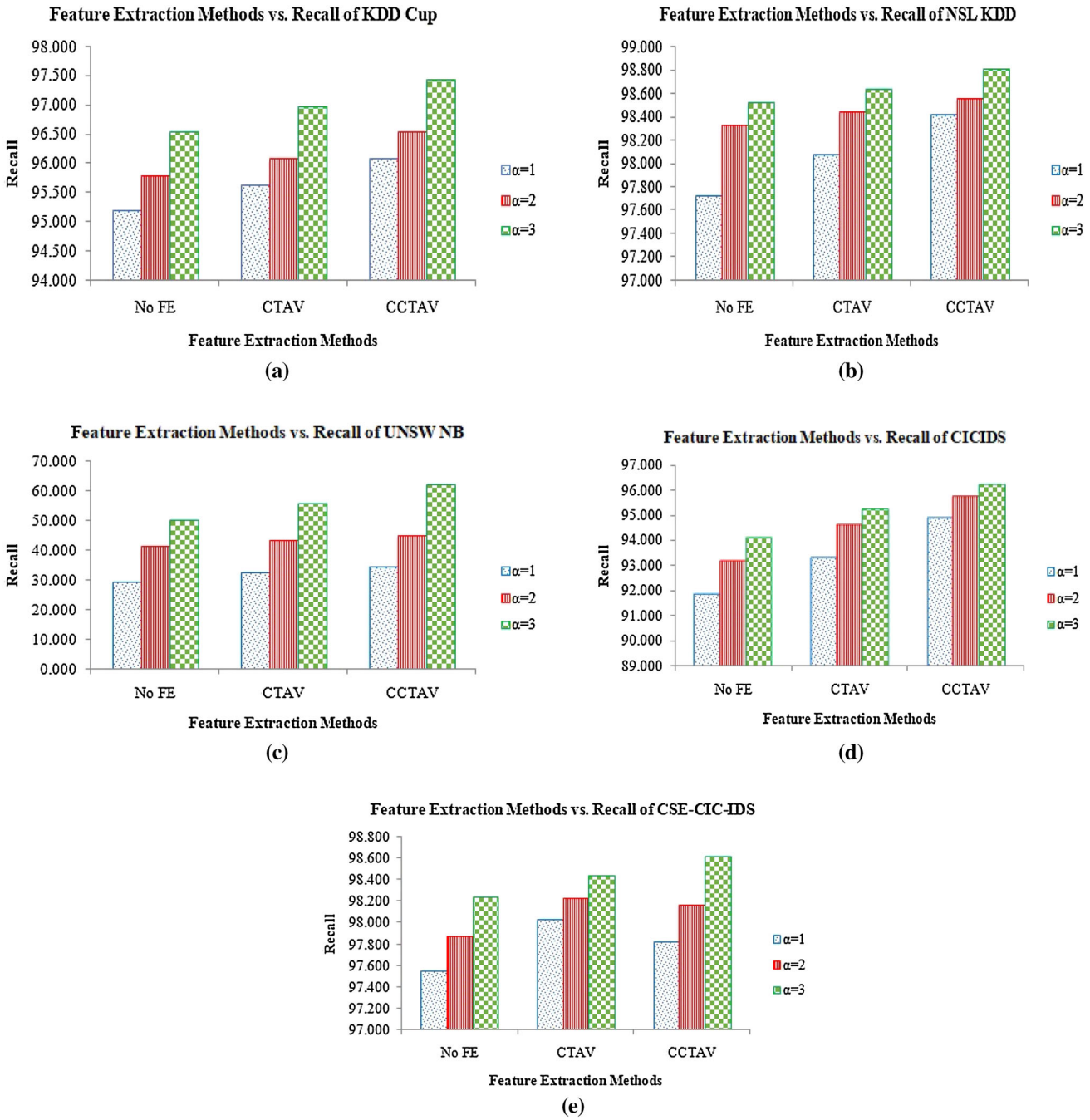

Fig. 3 Feature Extraction Methods vs. Recall. **a** KDD Cup, **b** NSL KDD, **c** UNSW NB, **d** CICIDS, **e** CSE-CIC-IDS

CICIDS, and CSE-CIC-IDS with varying thresholds. It is evident from Fig. 6a–e that the obtained accuracy is high for attack detection system with extracted features as it detects anomaly better compared to the statistical attack detection without FE and CTAV method.

Figure 7 depicts the FE methods versus ER for the five datasets, such as KDD Cup, NSL KDD, UNSW NB, CICIDS, and CSE-CIC-IDS with varying thresholds. It is evident from Fig. 7a–e, that the obtained ER is low for attack detection system with extracted features. It is observed that the ER is low for the proposed method as it detects anomaly better compared to the statistical attack detection without FE and CTAV method.

It is observed from the experiments that there is a reduction of 0.893% and 0.701% in FPR by the proposed CCTAV approach compared to statistical approach without FE and CTAV respectively for KDD Cup dataset. There is a reduction of 2.463% and 1.627% in FPR compared to

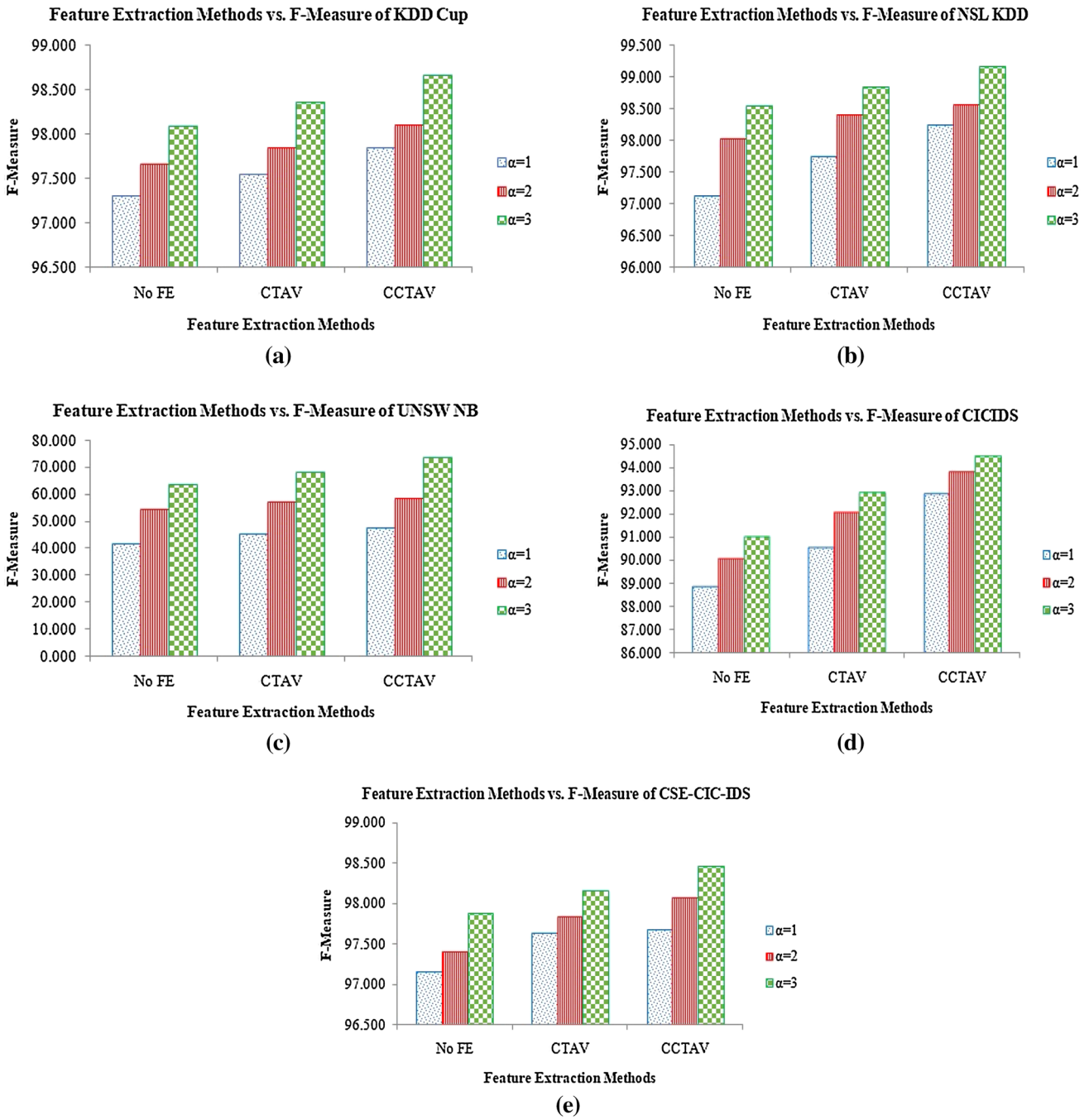

Fig. 4 Feature extraction methods vs. F-Measure. **a** KDD Cup, **b** NSL KDD, **c** UNSW NB, **d** CICIDS, **e** CSE-CIC-IDS

statistical approach without FE and CTAV respectively for NSL KDD dataset. There is a reduction of 0.743% and 0.347% in FPR compared to statistical approach without FE and CTAV respectively for UNSW NB dataset. There is a reduction of 0.477% and 0.231% in FPR compared to statistical approach without FE and CTAV respectively for CICIDS dataset. Also, there is a reduction of 0.979% and 0.420% in FPR compared to statistical approach without FE and CTAV respectively for CIC-CSE-IDS dataset.

Further, it is evident from the experiments that there is an improvement of 1.245% and 0.855% in accuracy by the proposed CCTAV approach compared to statistical approach without FE and CTAV respectively for KDD Cup dataset. There is an improvement of 1.164% and 0.765% in accuracy compared to statistical approach without FE and CTAV respectively for NSL KDD dataset. There is an improvement of 5.588% and 3.365% in accuracy compared to statistical approach without FE and CTAV respectively

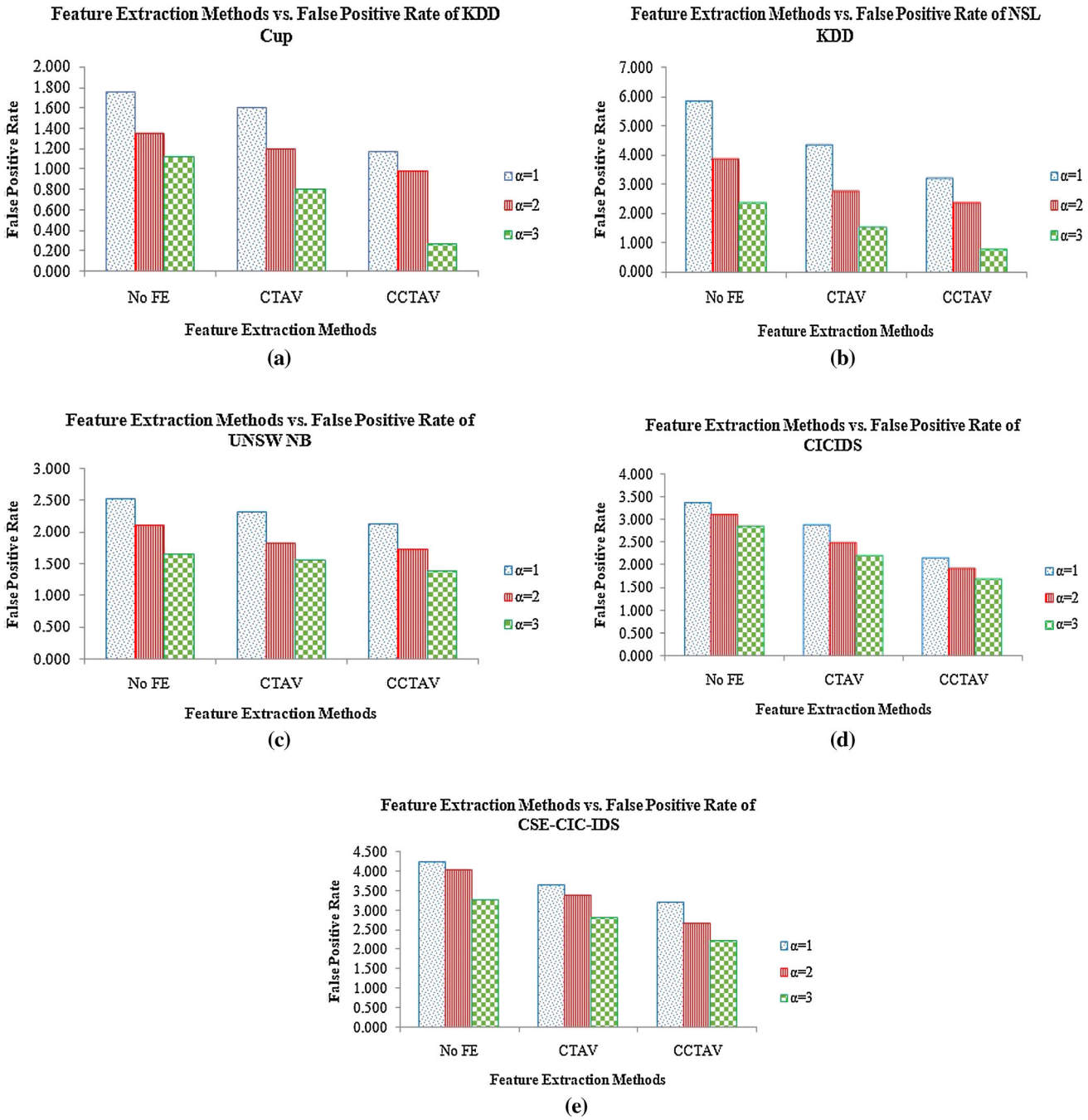

Fig. 5 Feature extraction methods vs. false positive rate. **a** KDD Cup, **b** NSL KDD, **c** UNSW NB, **d** CICIDS, **e** CSE-CIC-IDS

for UNSW NB dataset. There is an improvement of 0.633% and 0.271% in accuracy compared to statistical approach without FE and CTAV respectively for CICIDS dataset. Also, there is an improvement of 0.876% and 0.440% in accuracy compared to statistical approach without FE and CTAV respectively for CIC-CSE-IDS dataset. In all the cases, $\alpha = 3$ exhibits promising results as 99.7% of the records are within $(-3, +3)$ interval of the normal distribution parameter.

4.4 Tenfold cross validation

Ten-fold cross validations were performed for KDD Cup, NSL KDD, UNSW NB, CICIDS, and CSE-CIC-IDS datasets to validate the performance of the proposed approach and the performance metrics, i.e., F-Measure, FPR, Accuracy, and ER were measured for each folds and the mean and standard deviation (SD) of the validations are tabulated in Table 4. It is evident that most of the results of

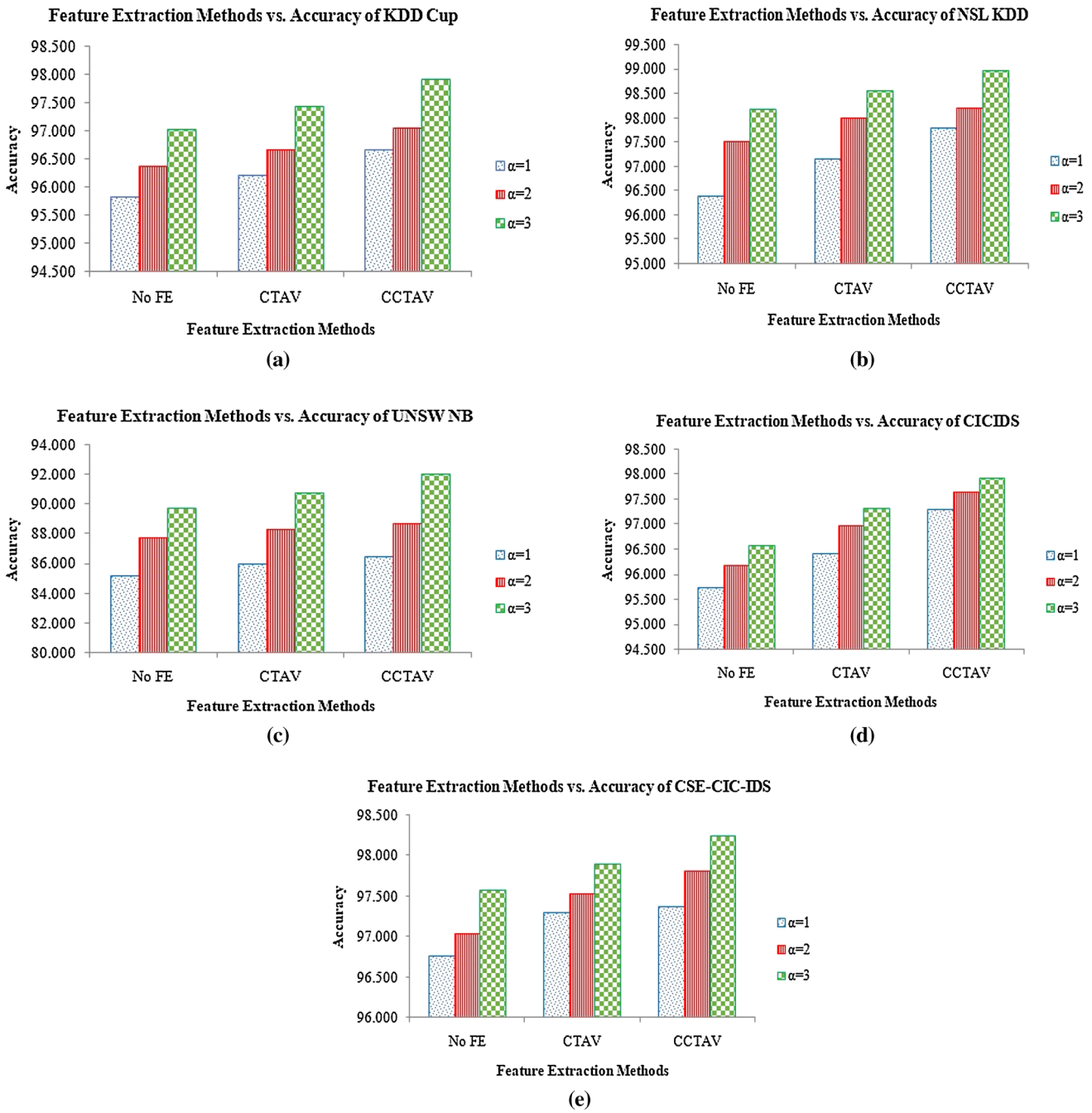

Fig. 6 Feature extraction methods vs. accuracy. **a** KDD Cup, **b** NSL KDD, **c** UNSW NB, **d** CICIDS, **e** CSE-CIC-IDS

the proposed approach are within the confidence interval (CI).

95% CI is computed to measure the reliability of the estimation of the proposed approach. The results obtained from each fold is averaged and the CI in terms of mean statistics, $CI_{95_{mean}}$ is computed using (21) and tabulated in Table 4.

$$CI_{95_{mean}} = Mean \pm \left(1.96 \times SD/\sqrt{N}\right) \tag{21}$$

where N is the number of folds. It is evident that the results obtained using the proposed approach are promising and almost within $CI_{95_{mean}}$.

Further, to handle the outliers effectively, the lower and upper 95% CI in terms of median statistics are computed using (22) and (23) respectively.

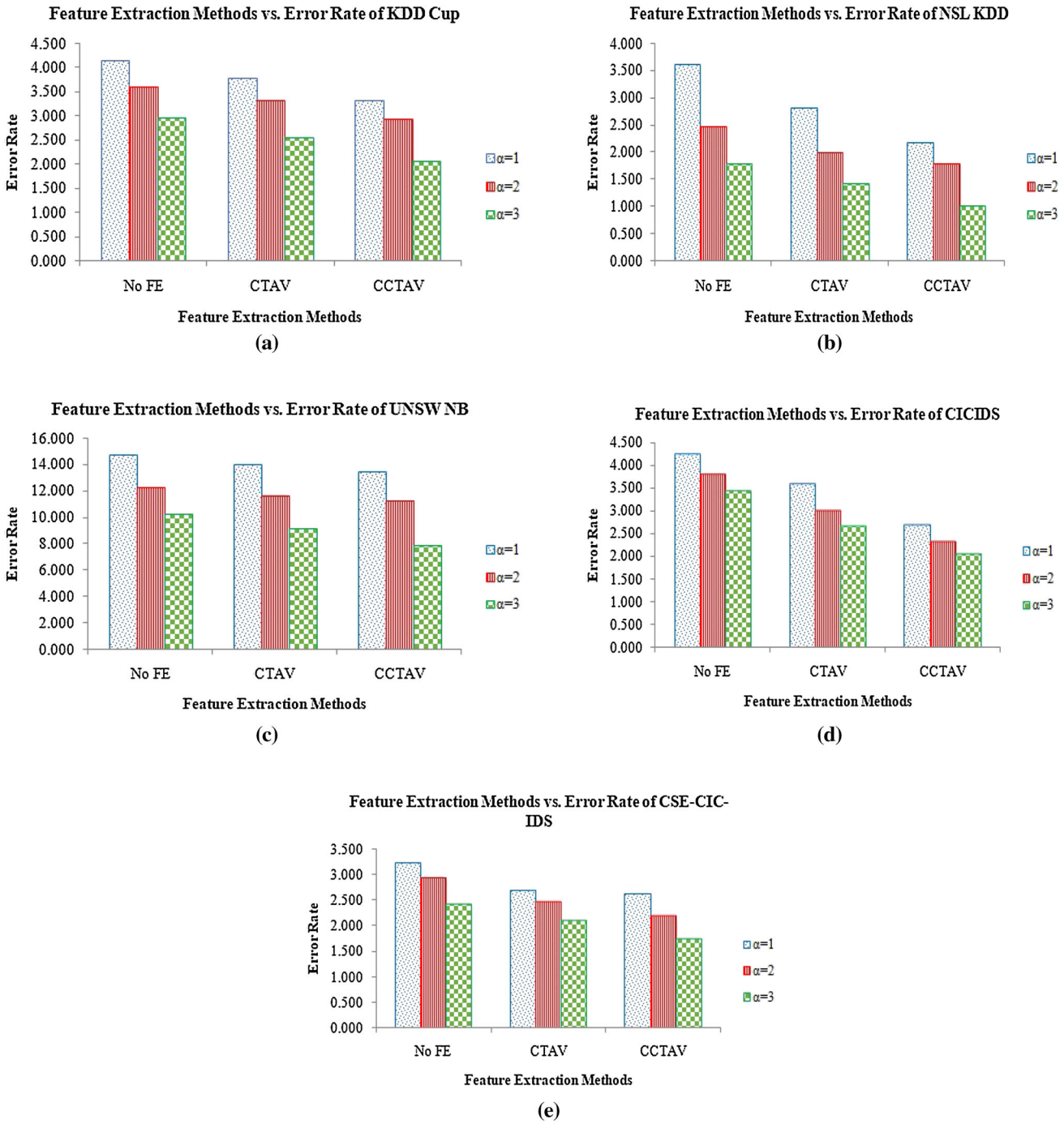

Fig. 7 Feature extraction methods vs. error rate. **a** KDD Cup, **b** NSL KDD, **c** UNSW NB, **d** CICIDS, **e** CSE-CIC-IDS

$$CI_{95md_{low}} = N/2 - 1.96 \times \sqrt{N}/2 \tag{22}$$

$$CI_{95md_{up}} = 1 + N/2 + 1.96 \times \sqrt{N}/2 \tag{23}$$

The $CI_{95md_{low}}$ and $CI_{95md_{up}}$ provide the ranked value i.e., the position and not the actual value. It is inferred from Table 4 that the outliers have been significantly detected by median statistics as the 95% CI range using the median statistics is wider compared to the mean statistics.

4.5 Time complexity analysis

The time complexity [39] of the proposed statistical CCTAV method is computed by considering the feature extraction method and statistical DoS attack detection method. The two processing steps involved for feature extraction are CC computation and TAV generation. The CC computation requires only mean computation of each

Table 4 95% confidence interval with mean and median statistics

Dataset	Thresh	Metrics	Mean	Median	SD	95% CI using Mean		95% CI using Median	
						From	To	From	To
KDD Cup	$\alpha = 1$	F-Measure	88.8151	98.3576	0.6751	88.3967	89.2335	97.3897	99.0269
		FPR	1.6906	1.4759	0.9747	1.0865	2.2947	0.9375	2.2677
		Accuracy	97.5041	97.4145	1.0489	96.8540	98.1542	95.9365	98.4583
		ER	2.4959	2.5855	1.0489	1.8458	3.1460	1.5417	4.0635
	$\alpha = 2$	F-Measure	89.9008	98.6676	0.2143	89.7680	90.0336	97.5588	99.0937
		FPR	1.2789	1.0547	0.7996	0.7833	1.7741	0.4687	2.2677
		Accuracy	97.7417	97.8989	1.0050	97.1188	98.3646	96.1934	98.5663
		ER	2.2583	2.1012	1.0050	1.6354	2.8812	1.4337	3.8066
	$\alpha = 3$	F-Measure	89.8835	99.0832	0.2069	89.7553	90.0117	98.4874	99.5129
		FPR	0.4174	0.3452	0.2655	0.2529	0.5820	0.1520	0.6271
		Accuracy	98.3019	98.5496	0.9855	97.6911	98.9127	97.6220	99.2266
		ER	1.6981	1.4504	0.9855	1.0873	2.3089	0.7734	2.3780
NSL KDD	$\alpha = 1$	F-Measure	97.4805	97.2303	0.9713	96.8785	98.0825	96.3048	98.5485
		FPR	2.8012	2.7885	1.4104	1.9270	3.6753	0.8974	4.2949
		Accuracy	97.7042	97.4824	0.8979	97.1476	98.2607	96.6197	98.6967
		ER	2.2958	2.5176	0.8979	1.7393	2.8524	1.3033	3.3803
	$\alpha = 2$	F-Measure	98.1636	98.0583	0.5513	97.8219	98.5053	97.8108	98.9449
		FPR	1.8396	2.0186	0.5797	1.4803	2.1989	0.8974	2.3718
		Accuracy	98.3380	98.2395	0.4987	98.0290	98.6471	98.0282	99.0490
		ER	1.6620	1.7605	0.4987	1.3529	1.9711	0.9510	1.9718
	$\alpha = 3$	F-Measure	98.6658	99.7705	0.5215	98.3426	98.9890	97.9070	99.2194
		FPR	1.1410	0.9615	0.7364	0.6846	1.5974	0.4487	2.1154
		Accuracy	98.7957	98.8909	0.4741	98.5019	99.0896	98.0986	99.2958
		ER	1.2043	1.1092	0.4741	0.9104	1.4981	0.7042	1.9014
UNSW NB	$\alpha = 1$	F-Measure	72.1642	72.3289	2.9809	70.3167	74.0118	68.0638	75.5199
		FPR	5.7449	5.6260	1.5229	4.8010	6.6888	4.6524	7.9848
		Accuracy	90.2778	90.0772	1.1121	89.5886	90.9671	89.0050	91.2341
		ER	9.7222	9.7081	1.1121	9.0329	10.4114	8.7659	10.9950
	$\alpha = 2$	F-Measure	77.2533	76.5351	5.1493	74.0617	80.4449	72.4277	79.5394
		FPR	5.7057	5.6783	2.1173	4.3934	7.0180	3.5938	8.1809
		Accuracy	91.7101	91.7887	2.0527	90.4378	92.9824	89.7049	92.5372
		ER	8.2899	8.2113	2.0527	7.0176	9.5622	7.4628	10.2951
	$\alpha = 3$	F-Measure	79.1200	78.0887	4.7875	76.1527	82.0873	74.3116	83.7418
		FPR	4.3980	4.5939	1.1144	3.7082	5.0896	3.4109	5.5672
		Accuracy	92.6793	92.2464	1.6199	91.6753	93.6833	91.1480	94.1417
		ER	7.3207	7.7536	1.6199	6.3167	8.3247	5.8583	8.8520
CICIDS	$\alpha = 1$	F-Measure	92.0203	92.1447	0.3154	91.8248	92.2158	87.3103	95.3764
		FPR	3.3701	3.4229	1.6589	2.3419	4.3983	1.7400	5.2471
		Accuracy	96.8207	96.8615	1.4949	95.8942	97.7472	94.7772	98.2369
		ER	3.1793	3.1385	1.4949	2.2527	4.1058	1.7631	5.2228
	$\alpha = 2$	F-Measure	92.8383	92.9094	0.3154	92.6428	93.0337	89.6559	96.4974
		FPR	3.0694	3.0450	1.8040	1.9513	4.1875	1.1962	5.0677
		Accuracy	97.1481	97.2159	1.5164	96.2082	98.0880	95.7517	98.6799
		ER	2.8519	2.7841	1.5164	1.9120	3.7918	1.3201	4.2483
	$\alpha = 3$	F-Measure	94.6055	95.5847	0.3154	94.4100	94.8010	90.6416	97.0139
		FPR	2.2283	1.7019	1.2272	1.4676	2.9889	1.1962	4.1325
		Accuracy	97.8972	98.3190	1.0614	97.2393	98.5550	96.2523	98.8571
		ER	2.1028	1.6810	1.0614	1.4450	2.7607	1.1429	3.7477

Table 4 (continued)

Dataset	Thresh	Metrics	Mean	Median	SD	95% CI using Mean		95% CI using Median	
						From	To	From	To
CSE-CIC-IDS	$\alpha = 1$	F-Measure	97.0771	97.5317	1.5267	96.1308	98.0233	97.2675	97.7565
		FPR	0.9159	0.8085	0.6974	0.4836	1.3482	0.2661	1.5248
		Accuracy	96.7716	7.2513	1.6158	95.7701	97.7731	96.9480	97.5061
		ER	3.2284	2.7487	1.6158	2.2269	4.2299	2.4939	3.0520
	$\alpha = 2$	F-Measure	97.3968	97.7195	1.5513	96.4353	98.3583	97.3482	98.5320
		FPR	0.7112	0.4349	0.5307	0.3823	1.0402	0.2354	1.2998
		Accuracy	97.1260	97.4640	1.6382	96.1106	98.1414	97.0407	98.3566
		ER	2.8740	2.5360	1.6382	1.8586	3.8894	1.6434	2.9593
	$\alpha = 3$	F-Measure	98.1228	98.0413	0.6507	97.7195	98.5261	97.4104	98.6077
		FPR	0.8136	0.8085	0.5274	0.4867	1.1405	0.2661	1.2998
		Accuracy	97.9012	97.8073	0.7189	97.4556	98.3468	97.1117	98.4275
		ER	2.0988	2.1927	0.7189	1.6532	2.5443	1.5725	2.8883

group, which requires only $O(1)$ time. In the TAV generation, $e(e-1)/2$ triangle areas are formed and it takes e^2 processing steps, where e is the number of extracted features. Therefore, the time complexity for feature extraction of CCTAV is $O(1) + O(e^2)$, which works out to be $O(e^2)$. The profile construction of the proposed method uses MahD measure which comprises of mean, standard deviation, and covariance matrix computations each of which requires 1, n^2, and n^3 processing steps respectively. Therefore, the time complexity for profile construction is $O(n^3)$ with respect to the number of records. But in the proposed method the profile is constructed with respect to the number of extracted features and the time complexity works out to be $O\left((e^2)^3\right)$. The statistical attack detection uses MahD measure using the constructed profile and the time complexity is $O(n^2)$ which works out to be $O\left((e^2)^2\right)$. Therefore, the total time complexity of the proposed method works out to be $O\left((e^2)^3\right)$.

The proposed statistical CCTAV DoS attack detection method is compared with two existing DoS attack detection systems such as TAM based Multivariate Correlation Analysis (MCA) [8] and MCA based computer vision (CV) [29]. The reason for choosing these methods for comparison is that these methods are for DoS attack detection. Also, these methods used feature extraction before DoS attack detection.

In the TAM based MCA method, the cluster centers are computed using k-means clustering with the time complexity of $O(n)$ and the TAM generates $e(e-1)/2$ triangle areas which takes e^2 processing steps. Therefore, the time complexity for feature extraction works out to be $O(n+e^2)$. The normal profile generation uses MahD measure with the complexity of $O(n^3)$ and the time complexity with respect to the extracted features works out to be $O\left((n+e^2)^3\right)$. The attack detection uses MahD measure using the generated normal profile and the time complexity is $O(n^2)$ which works out to be $O\left((n+e^2)^2\right)$. Therefore, the total time complexity of the TAM based MCA method works out to be $O\left((n+e^2)^3\right)$.

In the MCA based CV method, the feature generation process is the same as that of the feature extraction process of TAM based MCA method and the time complexity of feature generation process is $O(n+e^2)$. The generated features are converted to images and the dimension is reduced using principal component analysis which requires n^3 processing steps and the time complexity works out to be $O\left((n+e^2)^3\right)$ with respect to the number of extracted features. The normal profile generation uses EMD measure with the complexity of $O(n^2)$ and the time complexity with respect to the extracted features works out to be $O\left(\left((n+e^2)^3\right)^2\right)$. The attack detection also uses EMD measure using the generated normal profile and the time complexity is $O(n^2)$ which works out to be $O\left(\left((n+e^2)^3\right)^2\right)$. Therefore, the total time complexity of the MCA based CV method works out to be $O\left(\left((n+e^2)^3\right)^2\right)$. Table 5 shows the comparison of time complexity of the proposed method with the existing DoS attack detection methods. From the time complexity analysis, it is evident that the time complexity of the proposed method is less compared to the existing methods.

Table 5 Comparison of time complexity

Approach	Process	Time complexity	Total time complexity
Proposed statistical CCTAV	CC computation	$O(1)$	$O\left((e^2)^3\right)$
	TAV generation	$O(e^2)$	
	Profile construction	$O(n^3) = O\left((e^2)^3\right)$	
	Attack detection	$O(n^2) = O\left((e^2)^2\right)$	
TAM based MCA [8] 2014	Cluster centers	$O(n)$	$O\left((n+e^2)^3\right)$
	TAM generation	$O(e^2)$	
	Normal profile generation	$O(n^3) = O\left((n+e^2)^3\right)$	
	Attack detection	$O(n^2) = O\left((n+e^2)^2\right)$	
MCA based CV [29] 2015	Cluster centers	$O(1)$	$O\left(\left((n+e^2)^3\right)^2\right)$
	Feature generation	$O(n) + O(e^2) = O(n+e^2)$	
	Dimensionality reduction	$O(n^3) = O\left((n+e^2)^3\right)$	
	Attack detection	$O(n^2) = O\left(\left((n+e^2)^3\right)^2\right)$	

4.6 State-of-the-art comparisons

The proposed statistical CCTAV based detection method is compared with existing state-of-the-art attack detection systems for KDD Cup, NSL KDD, and UNSW NB datasets and the reported results are tabulated in Table 6. It is also tested with CICIDS and CIC-CSE-IDS datasets but not compared with state-of-the-art methods as these datasets have been generated recently and only few approaches have been tested with these datasets. It is evident that the FPR and accuracy of the proposed method are promising compared to the state-of-the-art statistical attack detection methods for all the datasets. In comparison with [17], the accuracy is high but the FPR is also high for the proposed approach which could be mainly attributed to the consideration of both decreased FPR and increased accuracy for normal and DoS attacks traffic only and not for other attacks in the dataset which is not the case in [17]. The machine learning approaches such as [41–43] performed significantly better compared to the proposed statistical approach as these approaches took more time in constructing the learned model which is not the case in the proposed statistical CCTAV method. The advantage of the proposed method lies in constructing the model using only normal traffic records that requires lesser time compared to

Table 6 Performance metrics comparison of proposed method with state-of-the-art attack detection methods

Approach	KDD cup		NSL KDD		UNSW NB	
	FPR (%)	Accuracy (%)	FPR (%)	Accuracy (%)	FPR (%)	Accuracy (%)
TAM based MCA [8] 2014	1.25	99.93	–	–	–	–
DS-SVM [31] 2014	1.60	92.06	–	–	–	–
TAM based CV [29] 2015	0.58	93.50	–	–	–	–
CANN [16] 2015	2.95	99.28	–	–	–	–
Self-taught learning [40] 2016	–	–	0.45	97.40	–	–
GAA based ADS [17] 2017	–	–	0.4	98.1	5.8	91.8
MVO ANN [41] 2018	–	–	3.2	98.21	0.4	99.61
Semi-supervised learning [42] 2018	–	–	0.28	98.23	0.46	93.71
DeeRaI-CuI [43] 2019	–	–	0.28	99.69	3.18	96.15
Proposed statistical CCTAV	0.271	97.916	0.743	98.975	1.391	92.065

training both normal and attack records for constructing machine learning based model.

5 Conclusion

In this paper, class center based triangle area vector method is proposed for feature extraction and MahD is used to detect DoS attacks. The main objective of this paper is to detect the DoS attacks with high accuracy and low false alarm rate. The CCs were computed to generate TAV by reducing the computational cost involved in existing clustering approaches. The MahD was calculated for the generated TAV and based on the threshold the traffic is classified as either normal or attack. Experiments were conducted on the benchmark datasets such as KDD Cup, NSL KDD, UNSW NB, CICIDS, and CSE-CIC-IDS. It is observed that the proposed statistical CCTAV based DoS attack detection system achieves accuracy of 97.916%, 98.975%, 92.065%, 97.928%, and 98.243% on KDD Cup, NSL KDD, UNSW NB, CICIDS, and CSE-CIC-IDS datasets respectively. In all the datasets, the threshold with the normal distribution parameter, $\alpha = 3$ gives the highest detection accuracy. The proposed approach is validated using ten-fold cross validations and 95% CI is computed along with mean and median statistics. It is evident that most of the results are within the CI. Further, from the time complexity analysis, the computational complexity of the proposed statistical CCTAV method is seen to be less compared to the existing methods. The extension of this work could be the reduction of extracted features dimension, as the proposed method increases the dimension based on the number of attack classes.

Appendix

Illustration of proposed method

This section illustrates the proposed method. For illustrative purpose, only six features as tabulated in Table 7 [44] have been used. The sample data used for illustration is tabulated in Table 8. $R1$ to $R13$ represent the records, $F1$ to $F6$ represent the features, and Class denotes the target classes. The first step is to compute the CCs for the five classes of the sample data. The CCs for the five classes are

Table 8 Sample data

R. no.	*F1*	*F2*	*F3*	*F4*	*F5*	*F6*	Class
R_1	1	0.0033	0.0118	0	1	0	Normal
R_2	1	0.0039	0.0315	0.0366	1	0	Normal
R_3	1	0.0047	0.0295	0.2642	1	0	Normal
R_4	1	0.0039	0.0020	0.0894	1	0	Normal
R_5	1	0.0068	0.0020	0.2967	1	0	Normal
R_6	1	1	0	1	1	0	Back
R_7	1	1	0	1	1	0	Back
R_8	0.4	0	0.1102	1	0	0.8571	Neptune
R_9	0.4	0	0.1909	1	0.0104	1	Neptune
R_{10}	0	0.0189	1	1	1	0	Smurf
R_{11}	0	0.0189	1	1	1	0	Smurf
R_{12}	0.4	0.0005	0.0256	0.3171	0.1354	0.7143	Teardrop
R_{13}	0.4	0.0005	0.0453	0.3577	0.2292	0.5714	Teardrop

Table 7 Features used for Illustration

S. no.	Feature name	Value type	Description
1	service	Nominal	http, ftp, smtp, telnet, etc
2	src_bytes	Integer	Bytes sent in one connection
3	count	Integer	Sum of connections to the same destination IP address (Time traffic)
4	dst_host_count	Real	Sum of connections to the same destination IP address (Network traffic)
5	dst_host_same_srv_rate	Real	The percentage of connections that were to same services among the connections aggregated in dst_host_count
6	dst_host_diff_srv_rate	Real	The percentage of connections that were to different services among the connections aggregated in dst_host_count

Table 9 Class centers

Center	*F1*	*F2*	*F3*	*F4*	*F5*	*F6*
C_1	1	0.0045	0.0154	0.1374	1	0
C_2	1	1	0	1	1	0
C_3	0.4	0	0.1506	1	0.0052	0.9286
C_4	0	0.0189	1	1.1585	1	0
C_5	0.4	0.0005	0.0354	0.3374	0.1823	0.6429

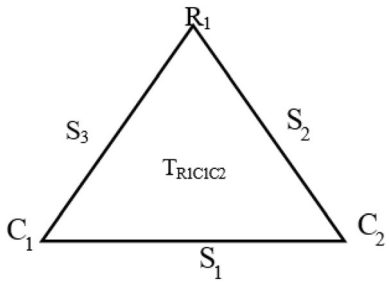

Fig. 8 Triangle $R_1C_1C_2$

computed and tabulated in Table 9. C_1, C_2, C_3, C_4, and C_5 are the CCs for the target classes such as Normal, Back, Neptune, Smurf, and Teardrop respectively. Once the CCs are known, then the TAV for each record is computed. The TAV is a vector that consists of 10 triangle areas. The TAV of record, R_1 is shown as follows:$[R_1C_1C_2, R_1C_1\ C_3, R_1C_1C_4, R_1C_1C_5, R_1C_2C_3, R_1C_2C_4, R_1C_2C_5, R_1C_3\ \ C_4, R_1\ C_3\ C_5, R_1C_4C_5]$.

The first triangle area of record, R_1 as depicted in Fig. 8, i.e., $R_1C_1C_2$ is computed. The triangle points of $R_1C_1C_2$, i.e., R_1, C_1, and C_2 are tabulated in Table 10. The points (C_1, C_2), (C_2, R_1), and (C_1, R_1) are for the sides of the triangle S_1, S_2, and S_3 respectively. The values obtained for three sides of the triangle S_1, S_2, and S_3 are 1.3173, 1.4119, and 0.1374 respectively. The perimeter of the triangle is 2.8666, the semi-perimeter of the triangle is 1.4333, and the triangle area obtained for $R_1C_1C_2$ is 0.0679. Similarly, the three sides of the triangle, perimeter of the triangle, semi-perimeter of the triangle, and the triangle area are obtained for the other 9 triangles. Then, the obtained TAV of R_1 is as follows: [0.0679 0.1025 1.1674 0.0952 1.1473 1.3352 0.0825 0.8647 0.3782 1.0170], which is shown in Fig. 9.

The mean of TAVs of normal traffic is computed and shown in Fig. 10. The profile is generated for detection using the mean TAV of normal records. The generated profile comprises of computed mean, -4.0175×10^8 and the standard deviation, 8.9723×10^8. The threshold, *Thresh* for the sample data is computed using the generated profile and tabulated in Table 11 for both the positive range

Table 10 Triangle points of $R_1C_1C_2$

	F1	*F2*	*F3*	*F4*	*F5*	*F6*
R_1	1	0.0033	0.0118	0	1	0
C_1	1	0.0045	0.0154	0.1374	1	0
C_2	1	1	0	1	1	0

	C_1	C_2	C_3	C_4
C_2	0.0679			
C_3	0.1025	1.1674		
C_4	0.0952	1.1473	1.3352	
C_5	0.0825	0.8647	0.3782	1.0170

Fig. 9 TAV of record, R_1

	C_1	C_2	C_3	C_4
C_2	0.0574			
C_3	0.1068	1.0771		
C_4	0.1025	1.0575	1.2630	
C_5	0.0829	0.8013	0.3951	0.9765

Fig. 10 Mean TAV of normal records

Table 11 Threshold computation

Threshold	Positive range	Negative range
$\alpha = 1$	4.9548×10^8	-1.2990×10^9
$\alpha = 2$	1.3927×10^9	-2.1962×10^9
$\alpha = 3$	2.2900×10^9	-3.0935×10^9

Table 12 Test cases

S. no.	*F1*	*F2*	*F3*	*F4*	*F5*	*F6*	Class
1	http	224	3	3	1	0	Normal
2	Private	28	30	102	0.29	0.04	Teardrop

and negative range. The computed profile is tested with two test cases and the test cases are tabulated in Table 12. The values of the test cases are normalized using min–max normalization and the normalized values of the test cases are shown in Table 13. The TAV of test case1 and test case2 are shown in Figs. 11 and 12 respectively. The MahD between TAV of test case1 and mean TAV of normal records is 1.1291×10^4 which lies within the *Thresh* and hence it is detected as *Normal*. The MahD between TAV of test case2 and mean TAV of normal records is -3.9192×10^9–1.9073×10^{-6}i which lies beyond the *Thresh* and hence it is detected as *Attack*.

Table 13 Normalized values of test cases

S. no.	*F1*	*F2*	*F3*	*F4*	*F5*	*F6*	Class
1	1	0.0041	0.0020	0	1	0	0
2	0.4	0.0005	0.0551	0.3929	0.2604	0.5714	1

	C_1	C_2	C_3	C_4
C_2	0.0689			
C_3	0.1021	1.1673		
C_4	0.0913	1.1473	1.3379	
C_5	0.0825	0.8645	0.3778	1.0184

Fig. 11 TAV of test case 1

	C_1	C_2	C_3	C_4
C_2	0.7433			
C_3	0.0000+0.5940i	0.6092		
C_4	0.0000+0.0507i	1.1630	0.5931	
C_5	0.0000+0.5940i	0.0724	0.0455	0.0533

Fig. 12 TAV of test case 2

References

1. Somani, G., Gaur, M.S., Sanghi, D., Conti, M., Buyya, R.: Ddos attacks in cloud computing: issues, taxonomy, and future directions. Comput. Commun. **107**, 30 (2017). https://doi.org/10.1016/j.comcom.2017.03.010
2. Velliangiri, S., Premalatha, J.: Intrusion detection of distributed denial of service attacks in cloud. Cluster Comput. (2017). https://doi.org/10.1007/s10586-017-1149-0
3. Yu, S., Tian, Y., Guo, S., Wu, D.O.: Can we beat ddos attacks in clouds? IEEE Trans. Parallel Distrib. Syst. **25**(9), 2245 (2014). https://doi.org/10.1109/TPDS.2013.181
4. Iot is a new backdoor for ddos attacks. https://www.grtcorp.com (2018).
5. P. Kasinathan, C. Pastrone, M.A. Spirito, M. Vinkovits, Denial of service detection in 6LoWPAN based Internet of things. In: 2013 IEEE 9th international conference on wireless and mobile computing, networking and communications (WiMob) (IEEE, 2013), pp. 600–607.
6. Akamai [State of the Internet]/Security q3 2017 Report. https://www.prnewswire.com (2017)
7. Hoque, N., Bhuyan, M.H., Baishya, R.C., Bhattacharyya, D.K., Kalita, J.K.: Network attacks: taxonomy, tools and systems. J. Netw. Comput. Appl. **40**, 307 (2014). https://doi.org/10.1016/j.jnca.2013.08.001
8. Tan, Z., Jamdagni, A., He, X., Nanda, P., Liu, R.P.: A system for denial-of-service attack detection based on multivariate correlation analysis. IEEE Trans. Parallel Distrib. Syst. **25**(2), 447 (2014). https://doi.org/10.1109/TPDS.2013.146
9. Derhab, A., Bouras, A.: Multivariate correlation analysis and geometric linear similarity for real-time intrusion detection systems. Secur. Commun. Netw. **8**(7), 1193 (2015). https://doi.org/10.1002/sec.1074
10. Weller-Fahy, D.J., Borghetti, B.J., Sodemann, A.A.: A survey of distance and similarity measures used within network intrusion anomaly detection. IEEE Commun. Surv. Tutor. **17**(1), 70 (2015). https://doi.org/10.1109/COMST.2014.2336610
11. Buczak, A.L., Guven, E.: A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun. Surv. Tutor. **18**(2), 1153 (2016). https://doi.org/10.1109/COMST.2015.2494502
12. Prasad, K.M., Reddy, A.R.M., Rao, K.V.: Defad: ensemble classifier for ddos enabled flood attack defense in distributed network environment. Cluster Comput. **21**(4), 1765 (2018). https://doi.org/10.1007/s10586-018-2808-5
13. Taheri, R., Javidan, R., Shojafar, M., Conti, M., et al.: Can machine learning model with static features be fooled: an adversarial machine learning approach. Cluster Comput. (2020). https://doi.org/10.1007/s10586-020-03083-5
14. Doriguzzi-Corin, R., Millar, S., Scott-Hayward, S., Martinez-del Rincon, J., Siracusa, D.: Lucid: a practical, lightweight deep learning solution for ddos attack detection. IEEE Trans. Netw. Serv. Manage. (2020). https://doi.org/10.1109/TNSM.2020.2971776
15. Tsai, C.F., Lin, C.Y.: A triangle area based nearest neighbors approach to intrusion detection. Pattern Recogn. **43**(1), 222 (2010). https://doi.org/10.1016/j.patcog.2009.05.017
16. Lin, W.C., Ke, S.W., Tsai, C.F.: Cann: an intrusion detection system based on combining cluster centers and nearest neighbors. Knowl. Based Syst. **78**, 13 (2015). https://doi.org/10.1016/j.knosys.2015.01.009
17. Moustafa, M., Slay, J., Creech, G.: Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks. IEEE Trans. Big Data (2017). https://doi.org/10.1109/TBDATA.2017.2715166
18. Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K.: Network anomaly detection: methods, systems and tools. IEEE Commun. Surv. Tutor. **16**(1), 303 (2014). https://doi.org/10.1109/SURV.2013.052213.00046
19. Zlomislic, V., Fertalj, K., Sruk, V.: Denial of service attacks, defences and research challenges. Cluster Comput. **20**(1), 661 (2017). https://doi.org/10.1007/s10586-017-0730-x
20. Zargar, S.T., Joshi, J., Tipper, D.: A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks. IEEE Commun. Surv. Tutor. **15**(4), 2046 (2013). https://doi.org/10.1109/SURV.2013.031413.00127
21. Somani, G., Gaur, M.S., Sanghi, D., Conti, M.: Ddos attacks in cloud computing: collateral damage to non-targets. Comput. Netw. **109**, 157 (2016). https://doi.org/10.1016/j.comnet.2016.03.022
22. Yan, Q., Yu, F.R., Gong, Q., Li, J.: Software-defined networking (sdn) and distributed denial of service (ddos) attacks in cloud computing environments: a survey, some research issues, and challenges. IEEE Commun. Surv. Tutor. **18**(1), 602 (2016). https://doi.org/10.1109/COMST.2015.2487361
23. Bharot, N., Verma, P., Sharma, S., Suraparaju, V.: Distributed denial-of-service attack detection and mitigation using feature selection and intensive care request processing unit. Arab. J. Sci. Eng. **43**(2), 959 (2018). https://doi.org/10.1007/s13369-017-2844-0
24. M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani: A detailed analysis of the KDD Cup 99 dataset. In Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on (IEEE, 2009), pp. 53–58. 10.1109/CISDA.2009.5356528
25. Iglesias, F., Zseby, T.: Analysis of network traffic features for anomaly detection. Mach. Learn. **101**(13), 59 (2015). https://doi.org/10.1007/s10994-014-5473-9
26. Moustafa, N., Slay, J.: The evaluation of network anomaly detection systems: statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set. Inf. Secur. J. **25**(13), 18 (2016). https://doi.org/10.1080/19393555.2015.1125974
27. Cicids 2017. https://www.unb.ca/cic/datasets/ids-2017.html (2017)

28. Cse-cic-ids 2018. https://www.unb.ca/cic/datasets/ids-2018.html (2018)
29. Tan, Z., Jamdagni, A., He, X., Nanda, P., Liu, R.P., Hu, J.: Detection of denial-of-service attacks based on computer vision techniques. IEEE Trans. Comput. **64**(9), 2519 (2015). https://doi.org/10.1109/TC.2014.2375218
30. Celebi, M.E., Kingravi, H.A., Vela, P.A.: A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Syst. Appl. **40**(1), 200 (2013). https://doi.org/10.1016/j.eswa.2012.07.021
31. Guo, C., Zhou, Y., Ping, Y., Zhang, Z., Liu, G., Yang, Y.: A distance sum-based hybrid method for intrusion detection. Appl. Intell. **40**(1), 178 (2014). https://doi.org/10.1007/s10489-013-0452-6
32. Jin, S., Yeung, D.S., Wang, X.: Network intrusion detection in covariance feature space. Pattern Recogn. **40**(8), 2185 (2007). https://doi.org/10.1016/j.patcog.2006.12.010
33. Deza, M.M., Deza, E.: Encyclopedia of Distances, pp. 1–583. Springer, New York (2009)
34. H.H. Chang, M.C. Lee, N. Chen, C.L. Chien, W.J. Lee: Feature extraction based Hellinger distance algorithm for non-intrusive aging load identification in residential buildings. In: Industry Applications Society Annual Meeting, 2015 IEEE (IEEE, 2015), pp. 1–8. 10.1109/IAS.2015.7356778
35. Z. Tan, A. Jamdagni, X. He, P. Nanda, R.P. Liu: Multivariate correlation analysis technique based on Euclidean distance map for network traffic characterization. In: International Conference on Information and Communications Security (Springer, 2011), pp. 388–398
36. Jamdagni, A., Tan, Z., He, X., Nanda, P., Liu, R.P.: Repids: a multi tier real-time payload-based intrusion detection system. Comput. Netw. **57**(3), 811 (2013). https://doi.org/10.1016/j.comnet.2012.10.002
37. Han, J., Pei, J., Kamber, M.: Data Mining: Concepts and Techniques. Elsevier, Amsterdam (2011)
38. I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In ICISSP (2018), pp. 108–116
39. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. MIT Press, Cambridge (2009)
40. A. Javaid, Q. Niyaz, W. Sun, M. Alam: A deep learning approach for network intrusion detection system. In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS) (2016), pp. 21–26
41. Benmessahel, I., Xie, K., Chellal, M.: A new evolutionary neural networks based on intrusion detection systems using multiverse optimization. Appl. Intell. **48**(8), 2315 (2018). https://doi.org/10.1007/s10489-017-1085-y
42. Idhammad, M., Afdel, K., Belouch, M.: Semi-supervised machine learning approach for ddos detection. Appl. Intell. **48**(10), 3193 (2018). https://doi.org/10.1007/s10489-018-1141-2
43. Ng, B.A., Selvakumar, S.: Deep radial intelligence with cumulative incarnation approach for detecting denial of service attacks. Neurocomputing **340**, 294 (2019). https://doi.org/10.1016/j.neucom.2019.02.047
44. Kdd cup features. https://www.aldapa.eus/res/README.pdf

N. G. Bhuvaneswari Amma received the M.E. degree in Computer Science and Engineering from College of Engineering, Guindy Campus, Anna University Chennai, India in 2009. She is currently working toward the Ph.D. degree in the Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, Tamil Nadu, India. Her areas of interest include computer networks, network security, machine learning, and statistical methods.

S. Selvakumar received the Ph.D. degree from the Indian Institute of Technology Madras (IITM), Chennai, India in 1999. He is the Professor in the Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, Tamil Nadu. He has been the Dean of IIIT, Tiruchirappalli, Tamil Nadu and currently the Director of IIIT, Una, Himachal Pradesh. He has to his credit of publishing 85 research papers. He was the investigator of Rs.100/- lakhs research project, Collaborative Directed Basic Research in Smart and Secure Environment (CDBR-SSE) Project sponsored by NTRO, Government of India. He is the investigator of Rs.106/- lakhs project, Nagarik Rog Pratirakshak: Unified Smart Immunization Coverage Monitoring and Analysis (UniSICMA) Project sponsored by Grand Challenges India—Immunization Data: Innovating for Action (IDIA). His research interests include network security, computer networks, high-speed networks, mobile networks, and wireless sensor networks. He is a member of the IEEE.