Ann. Data. Sci. (2015) 2(1):111–130
DOI 10.1007/s40745-015-0035-y

Novel Approach for Network Traffic Pattern Analysis using Clustering-based Collective Anomaly Detection

Mohiuddin Ahmed[1] · **Abdun Naser Mahmood**[1]

Received: 8 January 2015 / Revised: 23 April 2015 / Accepted: 24 April 2015 /
Published online: 14 May 2015

Abstract There is increasing interest in the data mining and network management communities in improving existing techniques for the prompt analysis of underlying traffic patterns. Anomaly detection is one such technique for detecting abnormalities in many different domains, such as computer network intrusion, gene expression analysis, financial fraud detection and many more. Clustering is a useful unsupervised method for both identifying underlying patterns in data and anomaly detection. However, existing clustering-based techniques have high false alarm rates and consider only individual data instances for anomaly detection. Interestingly, there are traffic flows which seem legitimate but are targeted at disrupting a normal computing environment, such as the Denial of Service (DoS) attack. The presence of such anomalous data instances explains the poor performances of existing clustering-based anomaly detection techniques. In this paper, we formulate the problem of detecting DoS attacks as a collective anomaly which is a pattern in the data when a group of similar data instances behave anomalously with respect to the entire dataset. We propose a framework for collective anomaly detection using a partitional clustering technique to detect anomalies based on an empirical analysis of an attack's characteristics. We validate our approach by comparing its results with those from existing techniques using benchmark datasets.

Keywords Traffic analysis · Computer security · Collective anomaly detection · Clustering

✉ Mohiuddin Ahmed
Mohiuddin.Ahmed@student.unsw.edu.au

Abdun Naser Mahmood
Abdun.Mahmood@unsw.edu.au

[1] School of Engineering and Information Technology, UNSW Canberra, Northcott Dr, Canberra, ACT 2600, Australia

1 Introduction

Network traffic analysis is a process of inferring patterns in communication which is a vital part of a network administrators job for maintaining the smooth operation of a network. Network professionals need to understand the underlying traffic patterns in data in order to protect a network and plan its capacity. In today's networked environment, the impact of even a limited period of network disruption is high for an organization. Consequently, there has been renewed interest in using a proactive and efficient network traffic analysis to detect anomalies in network traffic which is an important concern in terms of network security. The reliance on computer networks and their increasing connectivity has also raised the probability of damage being caused by various types of network attacks. These attacks, also called intrusions, are difficult to both detect and prevent, with security policies having to adjust to rapid changes in systems and applications. A threat/attack refers to anything which has detrimental characteristics which compromise a host or network. The poor design of a network, the carelessness of its users and/or the mis-configuration of its software or hardware cause vulnerabilities. Due to the continuous increase in variants of known network attacks, network traffic monitoring has become an important data analysis task [1]. It is almost impossible to predict the attacks an attacker is planning to launch. More importantly, a significant rise in sophisticated attacks has made the job of network professionals more challenging. To date, network attacks/intrusions have been handled by two dominant approaches: signature-based and anomaly detection. Although they seem totally opposite, they share a common characteristic or pitfall, that is, they depend on the knowledge supplied by a network expert.

Widely used signature-based systems [2] rely on extensive knowledge of the characteristics of network attacks. More clearly, a network expert provides details of the characteristics to the detection system so that an attack with a known pattern can be detected immediately it is launched. This is solely dependent on the attacks signature as a system is capable of detecting an attack only if its signature has been provided earlier by a network expert. It is clear that a system which can detect only what it knows is vulnerable when new attacks, which are constantly appearing in different versions and more stealthily, are launched. Even if a new attacks signature is created and incorporated in the system, the initial loss is irreplaceable and the repair procedure extremely expensive.

The concept of anomaly detection is to detect something which is different from the knowledge provided [3–7], with the system detecting everything not in its knowledge base as anomalous. More specifically, this genre of approaches relies on the normal traffic activity profile to build the knowledge base and consider activities which deviate from this baseline profile as anomalous. The advantage of anomaly-based over signature-based techniques is their capability to detect attacks which are completely novel, assuming that they exhibit ample deviations from the normal profile. Simultaneously, as normal traffic not included in the knowledge base is considered an attack, there will be inadvertent false alarms. Therefore, anomaly detection needs to be trained to build a normal activity profile which is time consuming and also depends on the availability of completely normal traffic datasets. In practice, it is very rare

and expensive to obtain attack-free traffic instances. Moreover, in today's dynamic and evolving network environments, it is really difficult to keep a normal profile up-to-date.

1.1 Motivation

The primary motivation of our research is to overcome the drawbacks of the above-mentioned two dominant approaches for network intrusion detection. In recent years, the traditional philosophy of using a knowledge base or external supervision has been superseded by unsupervised anomaly detection techniques which are based on purely fundamental aspects of data mining, such as clustering [8]. Without relying on expert supervision, unsupervised anomaly detection uses clustering techniques to determine the underlying structures of unlabeled data as well as unknown behaviors. These techniques for network traffic analysis make the following two assumptions about the data.

- **Assumption 1** 'The majority of the network connections are normal traffic, only a small percentage of traffic are malicious [9].
- **Assumption 2** 'The attack traffic is statistically different from normal traffic' [10].

The goal of unsupervised anomaly detection is to take a set of unlabeled data as input and accurately detect the attacks hidden in it. There is a possibility that the abovementioned assumptions will not always be true and, if not, the performance of the anomaly detection algorithm will be jeopardized; for example, DoS attack traffic instances do not follow these assumptions [11]. As DoS attacks are not significantly different from normal traffic and also occur in a similar fashion, their behaviors lead to the following assumption.

- **Assumption 3** The variance of traffic features of normal traffic is higher than the traffic features of attack traffic.

In this paper, we propose an anomaly detection framework based on this assumption which, unlike traditional anomaly detection methods, includes a collective anomaly detection technique. In our approach, we choose a clustering algorithm which is a variant of the basic *k-means* algorithm [12]. We evaluate our technique using the widely accepted 1998 DARPA [13] and 1999 KDD Cup [14] datasets. The logic behind using these datasets is that they have been used in a large proportion of research on network traffic analysis [15–19]. We also use the *Kyoto* dataset [32], which contains a present-day network traffic and attacks, to prove that our approach performs well not only with relatively old but also current network infrastructures.

This paper is structured as follows. In Sect. 2, we briefly discuss the major network attacks. In Sect. 3, we explain the clustering techniques used in our approach. We examine different aspects of anomaly detection and related works on it in Sect. 4 before discussing our methodology in Sect. 5. We extensively describe the experimental results in Sect. 6 and conclude the paper in Sect. 7.

Table 1 Distributions of attack classes in 1999 KDD cup dataset [14]

Attack	Training dataset (% Age)	10 % of training dataset (% Age)	Test dataset (% Age)
DoS	79.28	79.24	73.9
Probe	0.84	0.83	1.34
U2R	0.001	0.01	0.02
R2L	0.023	0.23	5.26

2 Major Network Threats

According to Kendall et al [20], attacks can be classified in the four major categories discussed below.

1. ***DoS*** is a type of misuse of the rights to the resources of a network or host. These attacks are targeted at disrupting a normal computing environment and rendering its service unavailable. A simple example of a DoS attack is denying legitimate users access to a web service when the server is flooded with numerous connection requests [21].
2. ***Probe*** is used to gather information about a targeted network or host and, more formally, for reconnaissance purposes.
3. ***User to Root (U2R)*** is used when an attacker aims to gain illegal access to an administrative account to manipulate or abuse important resources.
4. ***Remote to Local (R2L)*** is used when an attacker wants to gain local access as a user of a targeted machine. Most commonly, the attacker tries a trial and error approach to guess the password through automated scripts, a/the brute force method, etc.

In this paper, we focus on identifying a DoS attack as it is evident from the characteristics of the aforementioned attacks that its traffic volume is significantly high; for example, in the 1999 KDD Cup benchmark intrusion detection dataset [14], the proportion of network flows attributed to DoS attacks is much higher than those of other types, such as R2L, U2R and Probe. Table 1 shows the distributions of the classes of attacks. Not surprisingly, due to the volume of DoS attacks, most unsupervised anomaly detection techniques do not perform well for them due to the high false positive rates they obtain. The key issue is their underlying assumption that normal traffic is predominantly greater than anomalies, i.e., anomalies are rare incidents. However, this assumption does not hold for DoS attacks and, because of their volumes and inherent characteristics, it is a challenge to segregate normal and anomalous traffic in an unsupervised manner.

The key contribution of this paper is its accurate identification of DoS and similar attacks using the concept of collective anomaly-based *x-means* [22] clustering, as discussed in detail in the next section.

3 Clustering Methodology

Due to advances in computing and the proliferation of data repositories, to extract potentially useful information from data, appropriate knowledge discovery approaches are required. Clustering refers to unsupervised learning algorithms which do not need pre-labeled data to extract rules for grouping similar data instances [8]. The clustering process results in a given dataset being partitioned differently based on the different criteria used. Although there are different types of clustering techniques, we discuss those (*k-means* and *x-means*) that have been used to detect anomalous network activities.

3.1 Basic *k-means* Algorithm

The *k-means* is a well known unsupervised clustering algorithm [12] also known to converge to a local minimum of the distortion measure, i.e., the Euclidean distance. In it, each object is assigned to precisely one k cluster. Given two different sets of clusters produced by *k-means*, that which has the lower sum of squared error (SSE)/SST is considered better. SSE (1) and SST (2) are formally defined as

$$SSE = \sum_{i=1}^{k} \sum_{C_i} dist(c_i, x)^2 \quad where \ \forall x \in C_i \tag{1}$$

$$SST = \sum dist(C_m, x)^2 \quad where \ \forall x \ in \ Dataset \ D \tag{2}$$

3.2 Why Use *x-means* Algorithm?

The *k-means* algorithm is a well-known and widely used unsupervised algorithm [23]. However, it suffers from a few major drawbacks, one of which is the number of clusters (k) required to be provided by the user. An approach for providing the value of k is to determine the value of the objective function, i.e., the SSE (1) and analyze its change based on different values of k. In Eq. (1), C_i is the i^{th} cluster and c_i its centroid. However, as using this process for network traffic analysis is not very efficient due to the large data size, we use a variant of the *k-means* algorithm (*x-means*) which overcomes this problem through the following basic steps.

***x-means* clustering**

1. Implement Improve-Params
2. Implement Improve-Structure
3. If K> K_{max}, stop and report best scoring model found during search,
4. else go to step 1.

Unlike the *k-means* algorithm, the user specifies a range of k in which the potential k can be present. We apply the heuristic of taking the square root value of half the data

size as the maximum range [24] with two the minimum. This is also called a rule of thumb for estimating the number of clusters (3).

$$k = \sqrt{\frac{n}{2}} \tag{3}$$

Next, the Improve-Params part of the algorithm runs conventional *k-means* starting with the minimum value of k and continues until its maximum value is reached.

$$BIC(M_j) = l_j(D) - \frac{p_j}{2} \times \log R \tag{4}$$

Simultaneously, the Improve-Structure operation determines whether and, if so where, new centroids should appear by splitting the centroids and calculating the values of the Bayesian information criterion (BIC). That is, the given dataset D is clustered using the *k-means*. Then, each cluster is split into two parts indicating that it has two centroids which are moved to distances proportional to the size of the region in the opposite direction. In each original clustered region, the *k-means* is applied with $k = 2$, and the BIC score calculated for $k = 1,2$. Using Eq. (4), if BIC(K = 1)>BIC(K = 2), the cluster retrieves its original centroid, otherwise it retains the split version, where l_j(D) is the log-likelihood of the data according to the jth model and taken at the maximum likelihood point, p_j the number of parameters in M_j which refer to a family of alternative models and R the size of the dataset D.

4 Anomaly Detection

Anomaly detection is an important data analysis task, the main objective of which is to detect anomalous or abnormal data from a given dataset. It is an interesting area of data mining research as it involves discovering new and rare patterns in a dataset. It has been widely studied in statistics and machine learning, and is also known as outlier detection, novelty detection, deviation detection and exception mining [25–27].

4.1 Fundamental Aspects of Anomaly Detection

Based on the characteristics of data instances, anomalies are grouped in the following three categories.

- ***Point Anomaly*** refers to a particular data instance which deviates from the normal pattern of the dataset.
- ***Contextual Anomaly*** refers to a data instance which behaves anomalously in a particular context but not in others and is also termed a conditional anomaly.
- ***Collective Anomaly*** refers to a collection of similar data instances which behave anomalously with respect to the entire dataset. It might happen that an individual data instance is not an anomaly by itself but, due to its presence in a collection, is identified as an anomaly.

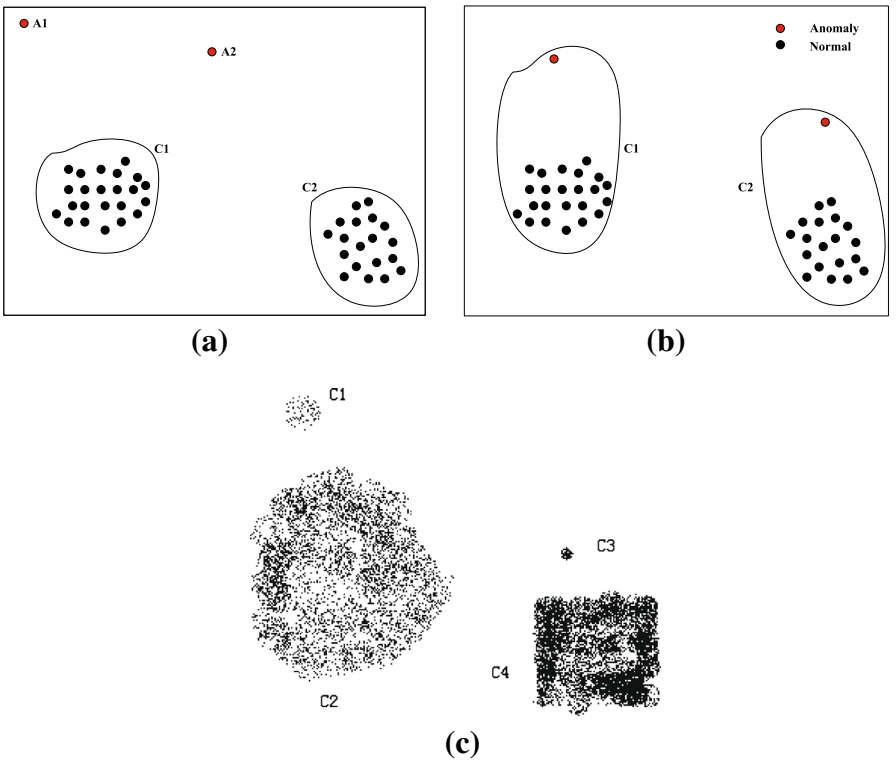

Fig. 1 Conceptual view of proposed approach, **a** Premise 1, **b** Premise 2, **c** Premise 3, adapted from [29]

4.2 Key Assumptions in Clustering-based Anomaly Detection/related Works

Since the goal of clustering is to group similar data, it can be used to detect anomalous patterns in a dataset. The following three key assumptions are made when using clustering to detect anomalies.

1. ***Premise 1*** if we create clusters of only normal data, any subsequent new data that do not fit well with the existing clusters are considered anomalies; for example, as density-based clustering algorithms do not include noise inside the clusters, noise is considered anomalous. In Fig. 1a, C1 and C2 are clusters containing normal instances, and A1 and A2 anomalies.
2. ***Premise 2*** it has been found that, when a cluster contains both normal and anomalous data, the normal data lie close to the nearest clusters centroid but the anomalies are far away from the centroids (Fig. 1b) [23].
   Mohiuddin et al. [23] considered an outlier according to a points distance from its centroid. If this distance is a fixed multiple of the mean distances of all other data points from the centroid, it is considered an outlier. Formally, '*an object (o) in a set of n objects is an outlier if the distance between it and its centroid is greater than p times the mean of the distances between that centroid and other objects*'.

They also showed that removing outliers from clusters can significantly improve the objective function of clustering.

3. ***Premise 3*** in a clustering which has clusters of various sizes, the smaller and sparser ones can be considered anomalous and the thicker ones normal. Instances belonging to clusters the sizes and/or densities of which are below a threshold are considered anomalous. Amer et al. [28] introduced the local density cluster-based outlier factor (LDCOF) which can be considered a variant of the CBLOF [29]. The LDCOF score (8) is calculated as the distance to the nearest large cluster divided by the average distances to that clusters center of the elements in it. This score is **A** when $p \in C_i \in \text{SC}$, where $C_j \in \text{LC}$ and **B** when $p \in C_i \in \text{LC}$.

$$distance_{avg}(C) = \frac{\sum_{i \in C} d(i, C)}{|C|} \tag{5}$$

$$A = \frac{min(d(p, C_j))}{distance_{avg}(C_j)} \tag{6}$$

$$B = \frac{d(p, C_i)}{distance_{avg}(C_i)} \tag{7}$$

$$LDCOF(p) = A \mid B; \tag{8}$$

He et al. [29] proposed a definition for cluster-based local anomalies which states that all the data points in a certain cluster are considered anomalies rather than single points, as shown in Fig. 1c in which clusters C1 and C3 are considered anomalous. They used some numeric parameters, i.e., α and β, to identify a small cluster (SC) and large cluster (LC). This clustering technique depends on these parameters but it is not clear how their values for different datasets can be determined. They used the SQUEEZER algorithm to cluster data as it achieves both a high quality of clustering and can handle high-dimensional data. Then, the FindCBLOF algorithm determines the outlier factor of each individual record in the dataset, with the CBLOF(t) for each record (t) calculated as

$$CBLOF(t) = \begin{cases} |C_i| * min(d(t, C_j)) \; where \; t \in C_i, C_i \in SC \\ \qquad\qquad and \; C_j \in LC \; for \; j = 1 \; to \; b \\ |C_i| * (d(t, C_i)) \; where \; t \in C_i \\ \qquad\qquad and \; C_i \in LC \end{cases} \tag{9}$$

5 Network Knowledge-Independent Collective Anomaly Detection (NKICAD)

In this section, we discuss in detail our proposed methodology for NKICAD, including its motivation, relationship between a DoS attack and collective anomaly, feature selection, proposed algorithm and variance-based DoS attack detection.

Table 2 Network traffic sample

SrcIP	DstIP	SrcPort	DstPort	Protocol	Payload length
62.52.69.78	172.16.113.84	80	27232	TCP	1460
62.52.69.78	172.16.113.84	80	27232	TCP	0
62.52.69.78	172.16.113.84	80	27232	TCP	243
62.52.69.78	172.16.113.84	80	27232	TCP	1024
62.52.69.78	172.16.113.84	80	27232	TCP	845
62.52.69.78	172.16.113.84	80	27232	TCP	465
62.52.69.78	172.16.113.84	80	27232	TCP	481
62.52.69.78	172.16.113.84	80	27232	TCP	1032
62.52.69.78	172.16.113.84	80	27232	TCP	22

5.1 Relationship Between DoS Attack and Collective Anomaly

As previously discussed, when a collection of similar data instances behave anomalously with respect to the entire dataset, it is termed a collective anomaly [30]. It is very likely that an individual data instance is not an anomaly by itself but, due to its occurrence in a group, is referred to as anomalous. Since we know that a DoS attack are intended to disrupt a normal computing environment, such as making resources unavailable, it may flood a web server with numerous anomalous connection requests but with only a single normal one. Let us present an example of a collective anomaly based on a DoS attack. Table 2 displays samples of network traffic instances in which an individual one (*<62.52.69.78,172.16.113.84,80,27232,TCP,1460>*) is a normal connection but the same one in Table 3 appears more than once and in a group. In this scenario, the group of records of *<62.52.69.78,172.16.113.84,80,27232,TCP,1460>* is called a collective anomaly as well as a DoS attack.

5.2 Feature Selection

Publicly available benchmark network traffic datasets include the 1998 DARPA [13] and 1999 KDD Cup [14] from the MIT Lincoln Laboratory which, interestingly, have different sets of features. To accurately identify malicious traffic instances, we need to utilize the features able to reflect the behaviors of traffic instances. Although, in the 1999 KDD Cup dataset, there are forty-one features, we emphasize those which could help to identify both attacks that target service vulnerability and DoS attacks. These forty-one features are categorized in four major classes: basic/intrinsic, time-based, host-based and content-based. Of them, time-based traffic features are constructed to specifically detect high-volume fast-rate DoS attacks based on the number of connections made to the same destination host or service in the past two seconds. Also, in the DARPA dataset, along with regular features, such as the source, destination IP addresses, protocol, etc., we focus on the payload length because, as shown in [31], it plays a vital role in network intrusion detection.

Table 3 Network traffic sample with collective anomaly

SrcIP	DstIP	SrcPort	DstPort	Protocol	Payload length
62.52.69.78	172.16.113.84	80	27232	TCP	1460
62.52.69.78	172.16.113.84	80	27232	TCP	1460
62.52.69.78	172.16.113.84	80	27232	TCP	1460
62.52.69.78	172.16.113.84	80	27232	TCP	1460
62.52.69.78	172.16.113.84	80	27232	TCP	1460
62.52.69.78	172.16.113.84	80	27232	TCP	1460
62.52.69.78	172.16.113.84	80	27232	TCP	1460
62.52.69.78	172.16.113.84	80	27232	TCP	0
62.52.69.78	172.16.113.84	80	27232	TCP	243
62.52.69.78	172.16.113.84	80	27232	TCP	1024
62.52.69.78	172.16.113.84	80	27232	TCP	845
62.52.69.78	172.16.113.84	80	27232	TCP	465
62.52.69.78	172.16.113.84	80	27232	TCP	481
62.52.69.78	172.16.113.84	80	27232	TCP	1032
62.52.69.78	172.16.113.84	80	27232	TCP	22

Proposed NKICAD approach

Begin
Step 1. Calculate size of dataset (D)
Step 2. Find range (R)
Step 3. Use *x-means*(D) = $C_1, C_2, ...C_n$
Step 4. Sort $|C_1| < |C_2| < ... < |C_n|$
Step 5. $TF = \sum$ (traffic features)
Step 6. **for** each cluster $i = 1{:}n$
Step 7. *x-means*(C_i) = $C_1, C_2, ...C_k$
Step 8. **for** each cluster $j = 1{:}k$
Step 9. (**Var**) = variance of cluster (C_j)
Step 10. **if Var**(C_j) $\neq$ **Var**(C_{j+1})
Step 11. ***CA(i) = MIN***(**Var**(C_j))
Step 12. **else** recluster C_i
Step 13. **else if** C_i = ***CA(i)***
Step 14. ***end if***
Step 15. **end**
Step 16. **end**
Step 17. **for** $i = 1{:}n$
Step 18. ***CA = CA(1)*** $\cup$ ***CA(2)*** $\cup$ **....** $\cup$ ***CA(n)***
Step 19. ***end***
End

5.3 NKICAD Algorithm

The proposed **NKICAD** algorithm first calculates the size of the dataset to provide input for the *x-means* algorithm to cluster the dataset. The heuristic used to determine the range (R) of the number of clusters in the dataset is discussed in Sect. 3.2.

Table 4 Notations used in proposed NKICAD algorithm

Symbol	Description
D	Dataset
R	Range of number of clusters in dataset
C_i	Number of clusters produced
TF	Sum of all traffic features in/of connection
Var	Variance
CA	Collective Anomaly
MIN	Minimum

Table 5 Sample cluster in which no further clustering possible

Srcbytes	Dst-bytes	Count	Srv-count
511	511	0	1032
511	511	0	1032
511	511	0	1032
511	511	0	1032
511	511	0	1032
511	511	0	1032
511	511	0	1032
511	511	0	1032
511	511	0	1032
511	511	0	1032

After clustering, we follow the assumption of clustering-based anomaly detection in a slightly modified way. Instead of nominating the smaller and sparser clusters as anomalous, we provide the priority for them to be anomalous and sort the clusters based on size. Then, the traffic features of the clusters produced are summed to make a single new traffic feature and each cluster is re-clustered, with the variances of the newly produced clusters calculated. The cluster with the minimum variance is considered a collective anomaly. If there is a situation in which some clusters produced have the same variance, re-clustering is performed and, if no more clustering is possible, the whole cluster is considered a collective anomaly. Table 5 shows such a scenario using the 1999 KDD Cup dataset. When its underlying data do not permit clustering, a collection is considered anomalous without its variance compared. Finally, all the candidate collective anomalies are grouped together and called an anomalous collection in a binary fashion instead of a score being given to each instance. Table 4 defines the notations used in our proposed NKICAD algorithm.

5.4 Variance-based DoS Attack Detection

In our proposed approach, the rationale behind re-clustering the dataset is to calculate its variances (10). As the first clustering takes all the traffic features into account and may not group devious DoS attacks, we re-cluster each of the clusters and, in this case,

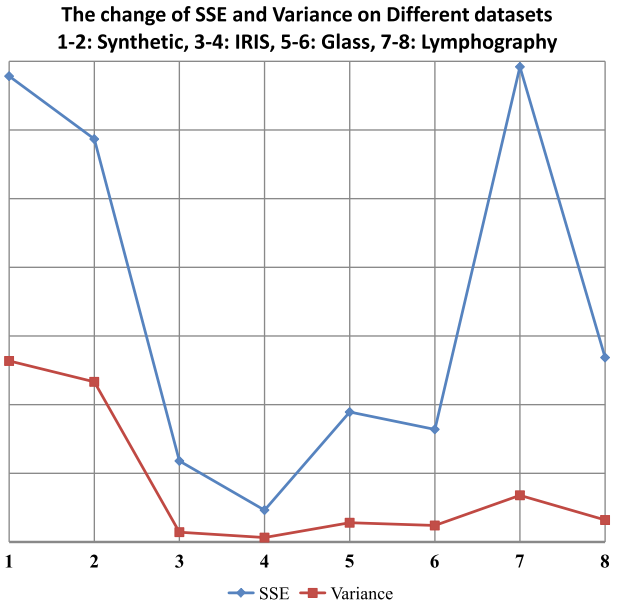

Fig. 2 Variance changes with SSE

summarize the traffic features by adding them together. This means that traffic features such as <*srcbyte, dstbyte, count and srvcount*> are converted to a new feature, the value of which is <*srcbyte* + *dstbyte* + *count* + *srvcount*>. Then, when clustering is applied again, a collective anomaly will be easier to detect by calculating the variances of the newly produced clusters; for example, the traffic sample in Table 5 will produce only one cluster and its variance will be zero. As previously stated, when a cluster cannot be re-clustered and the variance becomes zero, it is considered a collective anomaly.

$$Var(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2 \tag{10}$$

In Eq. (10), x_i is an instance and $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$ the mean of all instances. Instead of considering the SSE to differentiate the clustering results, we use variance because SSE is intended to measure clustering performances whereas variance is capable of identifying groups with similar traffic instances and we are converting the traffic features to a new one which is helpful for detecting a DoS attack. Interestingly, there is a relationship between SSE and variance. Mathematically, although they follow the same direction, variance is more appropriate for determining attack traffic following assumption 3 discussed in Sect. 1 whereas SSE helps to select the best clustering structure. We experimented with different datasets and found that, when the SSE decreased, so did the variance, as shown in Fig. 2.

However, in terms of network traffic analysis, we conclude that variance is more helpful for identifying attacks. When there is a DoS attack, the corresponding traffic

will have the same features but the normal traffic different ones. Therefore, the variance of attack traffic will be much lower than that of normal traffic.

6 Experimental Analysis

In this section, we validate our algorithm against the benchmark 1998 DARPA and 1999 KDD Cup datasets from the MIT Lincoln Laboratory [13,14]. Despite being criticized for having redundant records that prevent algorithms from learning infrequent harmful records, they are the only publicly available labeled benchmark datasets and the only ones widely used in the domain of intrusion detection research. Testing our approach on the KDD Cup 1999 data-set achieved a convincing evaluation and made comparisons with other state-of-the-art techniques equitable. In addition, the Kyoto dataset [32], which contains present-day network traffic and attacks, was also used to validate our approach. The experimental section provides the extensive research results obtained from our approach.

6.1 From Ocean to Pond

As, in order to examine the capability of our approach for identifying interesting patterns, we required network data containing known traffic patterns which are not readily available, we decided to use the widely accepted MIT Lincoln Laboratorys synthetic dataset for intrusion detection. This 1998 DARPA dataset provides 7 weeks of files containing raw traffic flow records, each with an associated label which indicates whether it was part of a normal or attack flow. In our experiment, we used traffic files from the Friday of the 7th week, with the rationale the presence of DoS attacks and their volumes. The overwhelming majority of the top ten attacks in this dataset is DoS ones [33] and, in particular, it includes *Neptune, Smurf* and *Back* attacks which fall into the DoS category. Due to the increasing bandwidths of networks, as a volume of network traffic data is often too large to analyze using traditional data mining techniques [34], sampling is a popular method for improving scalability. For our experiment, we used a subset of DARPA traffic, the Friday of the 7th week, containing top DoS attacks according to frequency. However, our future work will include investigating summarization techniques for application on network traffic. We used ipsumdump tool [35] to extract the raw data and converted them to a humanly readable format. For clustering, we converted the IP addresses into integers since integers can uniquely represent them and replaced the protocols with numbers, i.e., $TCP = 1$, $UDP = 2$, $ICMP = 3$. In the 1999 KDD Cup dataset, four major attack types are labeled, as discussed in Sect. 2. We used only the features capable of reflecting attack behaviors, i.e., basic and time-based ones. We took a set of DoS attack traffic and one of normal traffic from the original dataset in which all the features used are continuous.

6.2 Identification of Collective Anomaly

In this experiment, we used the sample 1999 KDD Cup dataset from the UCI machine Learning Repository [14] which has 4000 instances in the common class (normal) and

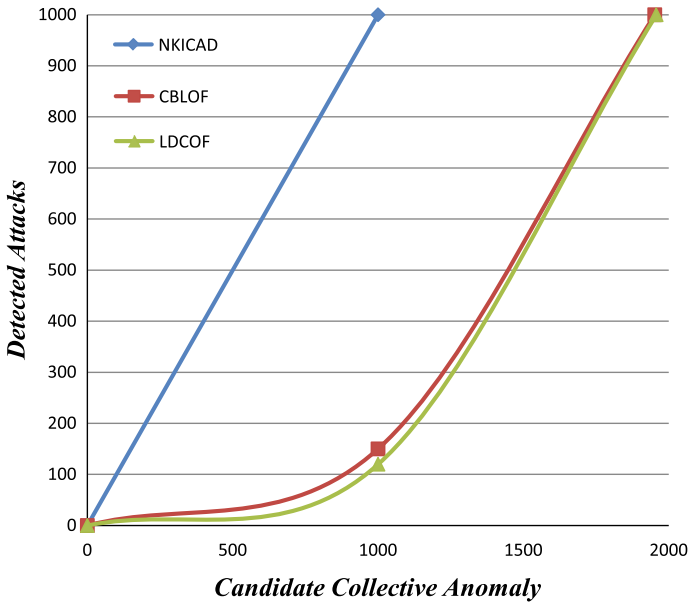

Fig. 3 Identification of attacks

1000 in the collective anomaly (DoS attack). The objective was to test whether the proposed anomaly detection technique could identify the rare class instances present. The probabilistic interpretation of the term *Recall* is that it is the probability that a relevant document is retrieved in a search. In the context of our experiment, we used the recall curve to show that the attacks detected from the benchmark dataset were retrieved correctly and, of the anomalies, the rare class instances were clearly identified.

Candidate collective anomalies have been compared against the top ratio anomalies in the CBLOF approach [29] and LDCOF method [28]. Top ratio anomalies are the number of anomalies specified as the *top-k* ones to the number of records in the dataset. Using the proposed technique, all the 1000 rare class instances in the dataset were found in only the first 1000 candidate collective anomalies, as shown in Fig. 3. This result clearly demonstrates that not only did the proposed method achieve 100 % recall by finding all the rare class instances but did so in fewer candidate anomalies than the CBLOF (1000 of 1952, 51.2 %) and LDCOF (1000 of 1957, 51 %).

6.3 Accuracy

We measured the accuracy of our approach using the standard confusion metrics true positive (TP), false positive (FP), true negative (TN) and false negative (FN) computed using equation (11).

- TP = no. of attacks correctly identified as attacks
- FP = no. of normal traffic incorrectly identified as attacks

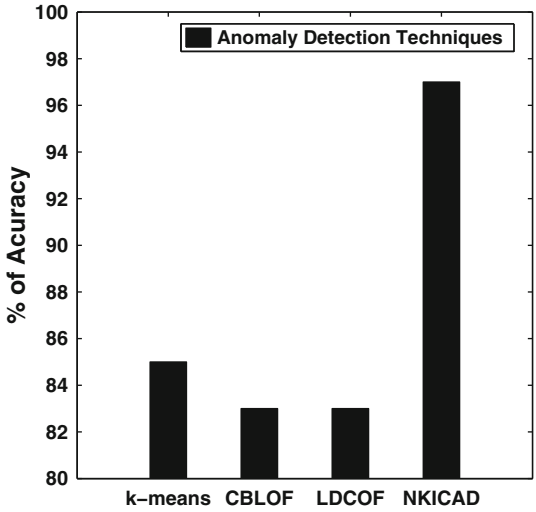

Fig. 4 Comparison of accuracies of NKICAD, *k-means*, CBLOF and LDCOF

Table 6 Comparison of attack detection of four/individual techniques

Attack	NKICAD	k-means	CBLOF	LDCOF
Neptune	√	√		
Smurf	√	√		
Back	√		√	√

– TN = no. of normal traffic correctly identified as normal
– FN = no. of attacks incorrectly identified as normal

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

The results from the DARPA dataset were compared with those from the CBLOF [29], LDCOF [28] and *k-means* algorithms, where clustering-based anomaly detection assumptions were used. Fig. 4 shows the experimental results which demonstrate that the proposed approach (NKICAD) (97 %) performed significantly better than the *k-means* (85 %), CBLOF (83 %) and LDCOF (83 %) techniques.

Table 6 also displays the capabilities of four techniques to detect the presence of attacks in the dataset used. Our approach could detect all three types whereas *k-means* detected two (*Neptune and Smurf*), and both CBLOF and LDCOF only one (*Back*).

6.4 Sensitivity

Sensitivity, also called the *true positive rate/hit rate/recall*, is another evaluation metric for anomaly detection techniques and is formally defined as

$$Hit\ Rate = \frac{TP}{TP + FN} \tag{12}$$

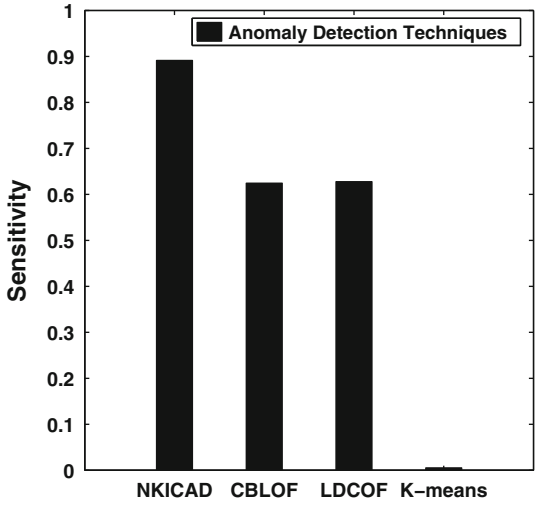

Fig. 5 Comparison of hit rates in Kyoto dataset [32]

For the Kyoto dataset[32], the hit rates for the clustering-based anomaly detection techniques and our proposed approach are shown in Fig. 5 in which it is evident that NKICAD outperformed the others.

6.5 Comparison of NKICAD and Random Selection

In this experiment, we tried to evaluate if our proposed approach identified anomalies by chance by comparing it with an algorithm for random selection which randomly selects the same number of anomalies. We used our approach to detect collective anomalies and check for attacks. Then, we randomly detected the same number of anomalies and checked for attacks in them. In the *Kyoto* dataset, our proposed approach detected 89.12 % of attacks whereas the random one found only 41.52 %. In the DARPA dataset, NKICAD detected 100 % of attacks but the random process only 66 %. Also, in the 1999 KDD Cup dataset, our approach could identify 100 % of attacks but the random process only 21 %. This comparison demonstrates the effectiveness of our approach for detecting DoS attacks, as summarized in Fig. 6.

Random detection

Begin
Step 1. Run NKICAD algorithm;.
Step 2. A = number of collective anomalies;
Step 3. Check for attacks in A;
Step 4. R = select A instances randomly from dataset; and
Step 5. Check for attacks in R.
End

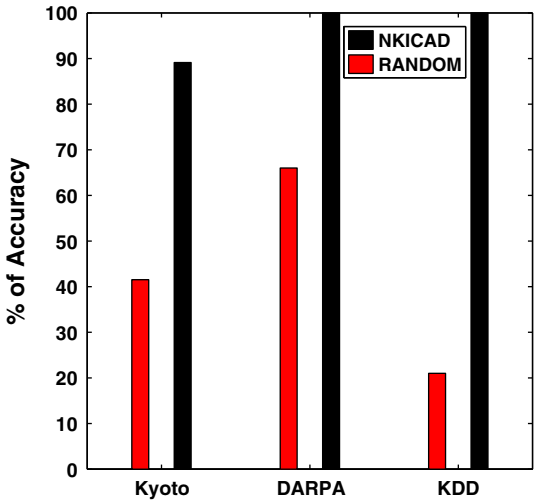

Fig. 6 Comparison of accuracies of NKICAD and random detection

6.6 Future Directions

One of the remaining challenges for network and data professionals is being able to analyze huge amounts of data in order to obtain knowledge. As computer network traffic is produced at a very fast rate, it is a difficult task to monitor a network in real time [36]. One promising solution to this problem can be using a summary of the data which can be analyzed faster and provide similar knowledge. Network traffic analysis can be performed on a summarized rather than complete dataset [37] and achieve higher efficiency in terms of attack detection. Therefore, summarization is a possible step in supporting real-time monitoring and reducing time complexity in network traffic analysis (Fig. 7). According to Hoplaros D. et al. [38], there are the following three important applications of data summarization in network traffic monitoring and intrusion detection.

- A summary of network traffic, instead of a huge dataset, can be used as input to anomaly detection algorithms to reduce the computational cost.
- A summary of network traffic can provide an overview of the activities of the network to the administrator.
- The alarms generated by intrusion detection systems can be summarized and ease the administrator's job.

Therefore, it is clear that the summarization of network traffic is an important aspect of research on both network traffic analysis and anomaly detection. In future, we will investigate data summarization and its use for network traffic analysis.

7 Conclusion

Anomaly detection is an important research area in the fields of data mining and machine learning. The current status of the increasingly complex and evolving internet

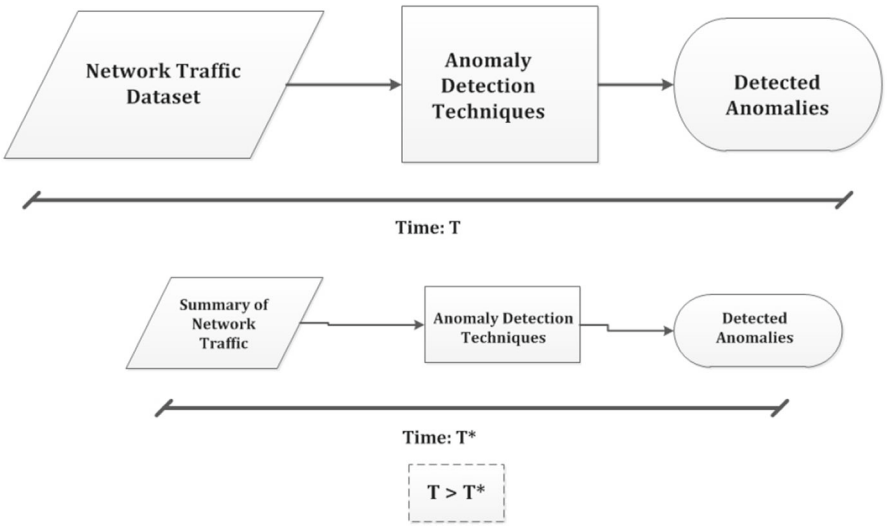

Fig. 7 Reduction in computational cost using anomaly detection techniques on summarized data

has raised the significance of efficient network traffic analysis. In this paper, we accentuated the limitations of existing approaches for network anomaly detection based on clustering and developed a framework for knowledge-independent network anomaly detection. Unlike traditional approaches of point anomaly, we incorporated the concept of a collective anomaly for accurately detecting the most dreaded network attacks such as denial of service (DoS). We utilized the strengths of clustering algorithms and the concept of variance to implement our proposal. To validate our approach, we used the 1998 DARPA, 1999 KDD Cup and *Kyoto* datasets for experimental analysis and demonstrated that our approach outperformed current clustering-based anomaly detection techniques. We also highlighted the importance of data summarization for network traffic analysis and, in future, will develop an efficient summarization technique for capturing the underlying distribution of data and creating a summary capable of monitoring high-capacity networks. We will also extend our approach to support online data streams for efficiently identifying interesting patterns.

8 Appendix

The variance is a measure of the dispersion of the random variable about the mean. The variance of a random variable X with mean μ is given (13) as follows:

$$\begin{aligned} var(X) &= \sigma^2 \\ &= E\left[(X - E(X))^2\right] \\ &= E\left[(X - \mu)^2\right] \end{aligned}$$

$$
\begin{aligned}
&= \int_{-\infty}^{\infty} (x-\mu)^2 f(x)dx \\
&= \int_{-\infty}^{\infty} x^2 f(x)dx - \left[\int_{-\infty}^{\infty} xf(x)dx\right]^2 \\
&= E\left(x^2\right) - E^2(x) \qquad (13)
\end{aligned}
$$

References

1. Hansman S, Hunt R (2005) A taxonomy of network and computer attacks. Comput Secur 24(1):31–43
2. Roesch M (1999) Snort—lightweight intrusion detection for networks. In: Proceedings of the 13th USENIX conference on system administration, LISA '99. USENIX Association, Berkeley, CA, USA, pp 229–238
3. Shi Y, Tian Y, Kou G, Peng Y, Li J (2011) Optimization based data mining: theory and applications. Springer, New York
4. Shi Y (2010) Multiple criteria optimization-based data mining methods and applications: a systematic survey. Knowl Inf Syst 24(3):369–391
5. Denning DE (1987) An intrusion-detection model. IEEE Trans Softw Eng 13(2):222–232
6. Thottan M, Ji C (2003) Anomaly detection in ip networks. IEEE Trans Signal Process 51(8):2191–2204
7. Barford P, Kline J, Plonka D, Ron A (2002) A signal analysis of network traffic anomalies. In: Proceedings of the 2Nd ACM SIGCOMM workshop on internet measurment, IMW '02. ACM, New York, NY, USA, pp 71–82
8. Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv 31(3):264–323
9. Portnoy L, Eskin E, Stolfo S (2001) Intrusion detection with unlabeled data using clustering. In: Proceedings of ACM CSS workshop on data mining applied to security (DMSA-2001, pp 5–8
10. Valdes A, Javitz HS (1993) The nides statistical component: Description and justification, In: Technical Report
11. Peng T, Leckie C, Ramamohanarao K (2002) Detecting distributed denial of service attacks using source ip address monitoring. In: Proceedings of the 3rd international IFIP-TC6 networking conference (Networking 2004, Springer, pp 771–782
12. MacQueen JB (1967) Some methods for classification and analysis of multivariate observations. In: Cam, LML Neyman J (Eds) Proceedings of the fifth berkeley symposium on mathematical statistics and probability, Vol. 1, University of California Press, pp 281–297
13. DARPA dataset, accessed: 2014–10-02.[Online]. Available: www.ll.mit.edu
14. KDD Cup dataset, accessed: 2014–10-02.[Online]. Available: www.kdd.ics.uci.edu
15. Leung K, Leckie C (2005) Unsupervised anomaly detection in network intrusion detection using clusters. In: Proceedings of the 28th Australasian conference on computer science—Volume 38, ACSC '05. Australian Computer Society Inc, Darlinghurst, Australia, Australia, pp 333–342
16. Brauckhoff D, Dimitropoulos X, Wagner et al (2009) Anomaly extraction in backbone networks using association rules. IEEE/ACM Trans Netw (TON) 20:1788–1799
17. Singhal A, Jajodia S (2006) Data warehousing and data mining techniques for intrusion detection systems. Distrib Parallel Databases 20(2):149–166
18. Ye N, Li X (2001) A scalable clustering technique for intrusion signature recognition. In: Proceedings of 2001 IEEE workshop on information assurance and security, pp 1–4
19. Gao M, Tian J, Xia M (2009) Intrusion detection method based on classify support vector machine. In: Intelligent computation technology and automation, 2009. ICICTA '09. Second international conference on, Vol. 2, 2009, pp 391–394
20. Kendall K (1999) A database of computer attacks for the evaluation of intrusion detection systems. In: DARPA off-line intrusion detection evaluation, proceedings of DARPA information survivality conference and eexposition (DISCEX), p 12–26
21. Ahmed M, Mahmood AN (2014) Network traffic pattern analysis using improved information-theoretic co-clustering based collective anomaly detection. In: Security and privacy in communication networks, lecture notes of the institute for computer sciences, social informatics and telecommunications engineering, Springer, Berlin Heidelberg

22. Dan Pelleg AM (2000) X-means: extending k-means with efficient estimation of the number of clusters. In: Proceedings of the 17th international conference on machine learning. Morgan Kaufmann, San Francisco, pp 727–734
23. Ahmed M, Naser A (2013) A novel approach for outlier detection and clustering improvement. In: Industrial electronics and applications (ICIEA), 2013 8th IEEE conference on, 2013, pp 577–582
24. Mardia KV, Kent JT, Bibby JM (1979) Multivariate analysis. Academic Press, London
25. Ahmed M, Mahmood AN, Hu J (2014) Outlier detection, In: The state of the art in intrusion prevention and detection, CRC Press, USA 2014, pp 3–23
26. Ahmed M, Mahmood AN, Islam MR (2015) A survey of anomaly detection techniques in financial domain. Futur Gener Comput Syst
27. Ahmed M, Anwar A, Mahmood AN, Shah Z, Maher MJ (2015) An investigation of performance analysis of anomaly detection techniques for big data in scada systems. EAI Endorsed Trans Ind Netw Intell Syst 2:2015
28. Mennatallah Amer MG (2012) Nearest-neighbor and clustering based anomaly detection algorithms for rapidminer, 1st edn. Shaker Verlag GmbH, Aachen
29. He Z, Xu X, Deng S (2003) Discovering cluster based local outliers. Pattern Recognit Lett 2003:9–10
30. Ahmed M, Mahmood A (2014) Network traffic analysis based on collective anomaly detection. In: Industrial electronics and applications (ICIEA), 2014 IEEE 9th Conference on, June 2014, pp 1141–1146
31. Wang K, Stolfo SJ (2004) Anomalous payload-based network intrusion detection. In: Jonsson E, Valdes A, Almgren M (Eds), RAID of lecture notes in computer science. Springer, New York, Vol. 3224, pp 203–222
32. Kyoto Dataset, accessed: 2014–10-02.[Online]. Available: www.takakura.com
33. Mahmood A, Leckie C, Udaya P (2007) A scalable sampling scheme for clustering in network traffic analysis. In: Proceedings of the 2nd international conference on scalable information systems, infoScale '07, 2007, pp 38:1–38:8
34. Claffy KC, Polyzos GC, Braun H-W (1993) Application of sampling methodologies to network traffic characterization. SIGCOMM Comput Commun Rev 23(4):194–203
35. Ipsumdump tool, accessed: 2014–10-02.[Online]. Available: www.cs.ucla.edu
36. Wang X, Abraham A, Smith KA (2005) Intelligent web traffic mining and analysis. J Netw Comput Appl 28(2):147–165
37. Zhu R (2011) Intelligent rate control for supporting real-time traffic in WLAN mesh networks. J Netw Comput Appl 34(5):1449–1458
38. Hoplaros D, Tari Z, Khalil I (2014) Data summarization for network traffic monitoring. J Netw Comput Appl 37:194–205