Network Traffic Analysis based on Collective Anomaly Detection

Mohiuddin Ahmed, Abdun Naser Mahmood, *Member, IEEE*
School of Engineering and Information Technology
University of New South Wales
Canberra, Australia
Email: mohiuddin.ahmed@student.unsw.edu.au, a.mahmood@unsw.edu.au

***Abstract*—There is a growing interest in the data mining and network management communities to improve the existing techniques for prompt analysis of underlying traffic patterns. Anomaly detection is one such technique to detect abnormalities in many different domains including computer network intrusion, gene expression analysis, financial fraud detection and many more. In this paper, we develop a framework to discover interesting traffic flows, which seem legitimate but are targeted to disrupt normal computing environment, such as Denial of Service attack. We propose a framework for collective anomaly detection using *x-means* clustering, which is a variant of basic *k-means* algorithm. We validate our approach by comparing against existing techniques and benchmark performance. Our experimental results are based on widely accepted DARPA dataset for intrusion detection from MIT Lincoln Laboratory.**

***Index Terms*—Network Traffic Analysis, Clustering, Anomaly Detection.**

I. Introduction

Network traffic analysis is a process to infer patterns in communication. It is vital part of a network administrators job to maintain smooth operation of their networks. Network professionals need to understand the underlying traffic pattern in order to plan network capacity and to protect their network. In today's networked environment, the impact of even a limited period of disruption is great for an organizations network. Consequently, there has been renewed interest in proactive and efficient network traffic analysis to detect anomalies in network traffic. Thus, efficient network traffic analysis is an important concern in network security.

Reliance on computer network and the increasing connectivity of these networks also raised the probability of damage caused by various types of network attacks. Network attacks, also named as intrusions are difficult to detect and prevent networks, with security policies due to the rapid change of system and applications. Simply, a threat/attack refers to anything which has the detrimental characteristics to compromise a host or network. Poor design of network, carelessness of users, misconfiguration of software or hardware cause the vulnerabilities. According to Kendall et al [1], the attacks can be classified into four major categories discussed below.

Denial of Service (DoS): It is a type of misuse of the rights to the resources of a network or a host. These are targeted to disrupt the normal computing environment and make the service unavailable. A simple example of DoS attack is denial of access to a web service to legitimate users, when the server is flooded with numerous connection requests. Performing DoS attack need no prior access to the target and thus considered to be a dreaded attack.
Probe: These attacks are used to gather information about a target network or host. More formally, these attacks are used for reconnaissance purpose. The reconnaissance attacks are quite common for gathering information about types and number of machines connected to a network, a host can be attacked to find out the types of softwares installed or application used. The probe attacks are considered as first step of an actual attack to compromise the host or network. There is no specific damage caused by these attacks but considered as serious threat to any corporation because it might give useful information to launch another dreadful attack.
User to Root: when the attacker aims to gain illegal access to administrative account to manipulate or abuse important resources, user to root attacks are launched. To launch such attacks, using social engineering approaches or sniffing password, the attacker access a normal user account. Then exploits some vulnerability to gain the super user privilege.
Remote to Local: When an attacker wants to gain local access as an user of a targeted machine, the R2L attacks are launched. The attacker have the privilege to send packets over network to the target machine. Most commonly the attacker tries hit and trial password guessing by automated scripts, brute force method etc. There are also some sophisticated attacks where attacker installs a sniffing tool to capture password before penetrating the system.

It is a research challenge to efficiently identify such attacks/intrusions in network traffic to prevent the network from probable damages. Anomaly detection is one such technique to identify abnormal behaviour and analyse further. Anomaly detection can be broadly classified into two categories which are supervised and unsupervised. Supervised learning is the machine learning task of inferring a function from labeled training data. Unsupervised learning tries to find hidden structure in unlabeled data, which distinguishes unsupervised learning from supervised learning. However, existing widely used network anomaly detection techniques rely on supervised learning. It requires the usage of labelled training data which is extremely expensive to produce. More importantly, these techniques are unable to detect zero day attacks due to the outrageous growth of cyber attacks and other vulnerabilities.

In this paper we focus on identifying denial of service attacks. It is evident from the characteristics of the aforementioned attacks that, traffic volume of DoS attack is significantly higher than other types of attacks. For example, in the 1999 KDD cup benchmark Intrusion Detection dataset [2], the proportion of network flows attributed to DoS attack is much higher than other types of attacks, such as R2L, U2R, or probe attacks. Table 1 shows the class distribution of the attacks. Not surprisingly, due to the volume of DoS attacks, unsupervised anomaly detection techniques do not perform well for DoS attacks due to high false positive rate. The key issue identified in the above problem is the underlying assumption in most unsupervised anomaly detection techniques that normal traffic is predominantly greater than anomalies, i.e., anomalies are rare incidents. However, in case of DoS attacks, this assumptions do not hold. Because of the volume of DoS attacks and inherent characteristics, it is a challenge to segregate normal and anomalous traffic in an unsupervised manner.

The key contribution of this paper is to accurately identify DoS attacks using the concept of collective anomaly based *x-means* [3] clustering by successfully identifying DoS and similar attacks as collective anomaly. The remainder of the paper is organized as follows. In Section 2, we briefly discuss different aspects of anomaly detection and clustering process followed by recent literature review in Section 3. Section 4 consists of a brief description of *x-means* algorithm and Section 5 includes the description of our proposed approach. In Section 6, we discuss the experimental results and conclude the paper in Section 7.

TABLE I
ATTACK CLASS DISTRIBUTION OF 1999 KDD CUP DATA [2]

Attack	Training Dataset (%age)	10% of TD (%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

TD: Training Dataset

II. ANOMALY DETECTION AND CLUSTERING

Anomalies are referred to as patterns in the data that do not conform to a well defined characteristic of normal patterns. Anomalies are generated due to variety of abnormal activities, e.g., credit card fraud, mobile phone fraud, cyber attacks etc. An important aspect is the nature of anomaly. Anomalies can be categorized in the following ways.

Point Anomaly: When a particular data instance deviates from the normal pattern of the dataset, it can be considered as a point anomaly. For a realistic example, we can consider expenditure on car fuel. If the usual fuel consumption per day is 5 litres, if some day it becomes 50 litres then obviously it is a point anomaly.
Contextual Anomaly: When a data instance is behaving anomalous in a particular context, but not in other times, then it is termed as a contextual anomaly, or conditional anomalies. For example, the expenditure on credit card during a festive period , e.g., Christmas or New Year, is usually higher than the rest of the year. Although, the expenditure during a festive month can be high, it may not be anomalous due to the expenses being contextually normal in nature. On the other hand, an equally high expense during a non-festive month could be deemed as a contextual anomaly.
Collective Anomaly: When a collection of similar data instances are behaving anomalously with respect to the entire data set, then this collection is termed as collective anomaly. It might happen that the individual data instance is not an anomaly by itself, but due to its presence in a collection it is identified as an anomaly. For example, in a human Electro-Cardio-Gram (ECG) output, the existence of low values for a long period of time indicates outlying phenomenon corresponding to abnormal premature contraction, however, one low value by itself is not considered as anomalous.

Next, we describe clustering, which is a popular technique to identify normal and anomalous data. Let us formally define clustering. Given a set of input patterns, X = x_1, x_2,.,x_N. Clustering attempts to seek a k-partition of X, C = C_1, C_2....C_k ($K \leq N$), such that
1. $C_i \neq \emptyset$, i =1,......,K; a cluster cannot be empty.
2. $\bigcup_{i=1}^{k} C_i$ = X; the clustering process should cluster all the data points.
3. $C_i \bigcap C_j$ = $\emptyset$, i,j = 1,.....,K and i $\neq$ j; each data object should belong to exclusively one cluster, however there are other clustering techniques e.g. fuzzy clustering, which extends the notion to associate each data object to more than one clustering using a membership function.
4. $d(x_i, x_j)$ in C_i is minimized; distance between points inside a cluster, the intra cluster distance is minimized.
5. $d(C_i, C_j)$ is maximized, distance between clusters, the inter-cluster distance is minimized.

A. Key Assumptions in Clustering based Anomaly Detection

Since the goal of clustering is to group together similar data, it can be used to detect anomalous patterns in a dataset. There are three key assumptions to use clustering as a way of detecting anomalous events. These are briefly discussed below.

Assumption 1: We can create clusters of normal data only, subsequently, any new data that do not fit well with existing clusters of normal data are considered as anomalies. For example, density based clustering algorithms do not include noise inside the clusters. Here noise is considered as anomalous. For example, in the Figure 1, C1 and C2 are clusters containing normal instances and A1, A2 are anomalies.

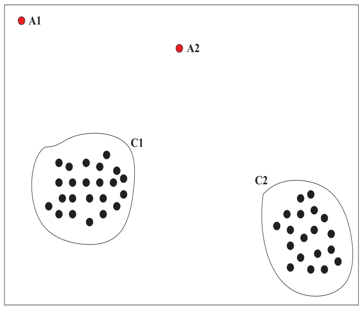

Fig. 1. Example of anomaly based on assumption 1

Assumption 2: When a cluster contains both normal and anomaly data, it has been found that normal data lie close to the nearest cluster centroid but anomalies are far away from the centroids [4]. Under this assumption, anomalous events are detected using a distance score.

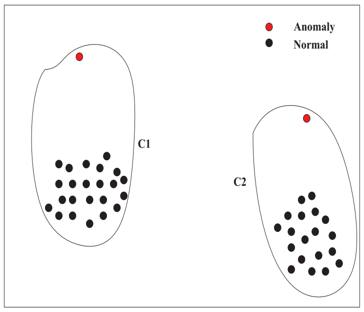

Fig. 2. Example of anomaly based on assumption 2

Assumption 3: In a clustering where there are clusters of various sizes, smaller and sparser can be considered as anomalous and thicker clusters can be considered normal. The instances belonging to clusters whose size and/or density is below a threshold are considered as anomalous.

III. Related Work

In this section, we briefly discuss the existing clustering based anomaly detection techniques. Interestingly, nearly all of the existing techniques detect anomalies based on analysis of individual data instances, even the data instances in small, sparse clusters are given individual anomaly scores.

Svetlona et al [5] presented an outlier removal clustering algorithm (ORC) that provides outlier detection and data clustering simultaneously. Their proposed algorithm has two stages. First, the *k-means* [6] clustering is applied and then *outlyingness factor*, o_i for each of the data point, x_i is calculated by taking the ratio of a point's distance to the centroid, C_{pi} and the maximum distance, d_{max} from the centroid to any other point, stated in equation (1). If outlying factor for any point is greater than a threshold T, it is considered as an outlier and removed from dataset. Their experimental data includes synthetic data and some map images. Mean Absolute Error (MAE) is used to evaluate their algorithm performance. The value of parameter T is dependent on the dataset, but how the value of T should be chosen for different types of dataset was not discussed.

$$o_i = \frac{\|x_i - C_{p_i}\|}{d_{max}} \tag{1}$$

He et al [7] proposed a definition for cluster based local anomalies. According to their definition, all the data points in a certain cluster are considered as anomalies rather than a single point, as shown in Figure 6. The clusters C1 and C3 are considered as anomalous. They used some numeric parameters, i.e. α, β to identify Small Cluster (SC) and Large Cluster (LC). The clustering technique depends on these parameters but it is not clear how the values can be determined for various datasets. They used the SQUEEZER algorithm to cluster data, as it achieves both high quality of clustering and can handle high dimensional data. Then the FindCBLOF algorithm determines outlier factor of each individual record in dataset. CBLOF(*t*) for each record *t* is calculated following equation (2):

$$CBLOF(t) = \begin{cases} |C_i| * min(d(t, C_j)) & where\ t \in C_i, C_i \in SC\ and \\ & C_j \in LC\ for\ j = 1\ to\ b \\ |C_i| * (d(t, C_i)) & where\ t \in C_i\ and\ C_i \in LC \end{cases} \tag{2}$$

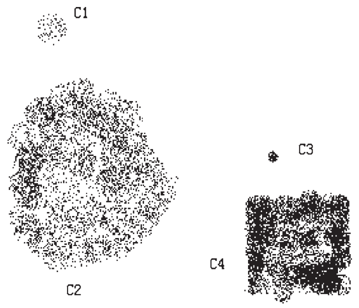

Fig. 3. Anomalous clusters C1,C3 [7]

Amer et al [8] introduced Local Density Cluster-Based Outlier Factor (LDCOF) which can be considered as a variant of CBLOF [7]. The LDCOF score (4) is calculated as the distance to the nearest large cluster divided by the average distance to the cluster center of the elements in that large cluster.

x-means Algorithm

```
1. Improve-Params.
2. Improve-Structure.
3. If K>$K_{max}$, Stop and report the best scoring model found during the search.
4. Else, Go to 1.
```

$$distance_{avg}(C) = \frac{\sum_{i \in C} d(i,C)}{|C|} \tag{3}$$

$$LDCOF(p) = \begin{cases} \frac{min(d(p,C_j))}{distance_{avg}(C_j)} \; if \; p \,\epsilon\, C_i \,\epsilon\, SC \; where \; C_j \,\epsilon\, LC \\ \frac{d(p,C_i)}{distance_{avg}(C_i)} \; if \; p \,\epsilon\, C_i \,\epsilon\, LC \end{cases} \tag{4}$$

IV. x-means Algorithm

k-means [6] algorithm is a well known and widely used unsupervised algorithm. However, it suffers from few major drawbacks and one of them is, the number of clusters *k* has to be provided by the user. An approach of providing the value of *k* is finding out the value of objective function i.e. Sum of Squared Error (SSE) (5) and analysing it's change based on different values of *k*. In equation (5), capital C_i is the i^{th} cluster and small c_i is the centroid of cluster C_i. However, using this process for network traffic analysis is not very efficient due to large data size. We use a variant of *k-means* algorithm, *x-means*, which overcomes the problem of *k-means* discussed above. *x-means* algorithm is displayed to show the basic steps.

$$SSE = \sum_{i=1}^{k} \sum_{C_i} dist(c_i, x)^2 \;\; where \;\; \forall x \in \; C_i \tag{5}$$

Unlike, *k-means* algorithm, the user specifies a range of *k*, where the potential *k* can be present. We apply the heuristic of taking the square root value of half of the data size as the maximum range [9] and two being minimum. It is also called as rule of thumb in order to estimate the number of clusters (6).

$$k = \sqrt{\frac{n}{2}} \tag{6}$$

Next, the **Improve-Params** part of the algorithm runs conventional *k-means* starting with the minimum value of *k* and continues until the maximum value of *k* is reached.
Simultaneously, the **Improve-Structure** operation finds out whether or where new centroids should appear, which is decided by splitting the centroids and calculating the Bayesian Information Criterion (BIC) value. For example, the given dataset D is clustered using *k-means*. Then each cluster is split into two parts, which indicates each cluster has got two centroids which are moved to a distance proportional to the size of the region in opposite direction. Then, in each original clustered region *k-means* is applied with *k*=2.

$$BIC(M_j) = l_j(D) - \frac{p_j}{2} \times \log R \tag{7}$$

Now BIC score is calculated for *k*=1,2. Using equation (7), if BIC(K=1)> BIC(K=2), the cluster retrieves its original centroid, else keeps the split version. where l_j (D) is the log-likelihood of the data according to the j_{th} model and taken at the maximum likelihood point, p_j is the number of parameters in M_j, which refers to a family of alternative models. *R* refers to the size of dataset *D*.

CAD Algorithm

```
Input: D(SrcIP,DstIP,Protocol,Payload Length), range of k
Output: n Clusters, Collective Anomaly Detected:CAD.
Begin
1. Perform x-means clustering on dataset D.
2. Sort the clusters based on size.
3. for each cluster i=1:n
4.        Apply x-means to cluster the payload length attribute.
5.        Calculate the variances (Var) of the produced clusters.
6.        CA(i)= cluster with min(Var).
7.        CAD(i) = <SrcIP,DstIP,Protocol,Payload Length> in CA(i).
8. end
9.    for i=1:n
10.    CAD = $\bigcup_{i=1}^{n}$ CAD(i)
11.    end
End
```

V. Collective Anomaly Detection (CAD)

This section explains how collective anomaly is identified using *x-means* algorithm. At first, the input data is extracted from network traffic as <*SrcIP, DstIP, Protocol, Payload Length*>, which is used for clustering. It has been shown that these four traffic attributes are sufficient to identify a DoS attack [10]. For example, a DoS attack can be denial of access to a web service to a legitimate user, when the service is flooded with numerous connection requests. Performing DoS attack needs no prior access to the target.

The key idea behind collective anomaly detection is that, a group of traffic flows collectively, with the same <*SrcIP, DstIP, Protocol, Payload Length*> pose the high probability of being DoS attack. The basic assumption of anomaly detection using traffic analysis is that, *'the variance of payload length of normal traffic flow is higher than the variance of payload length of attack, such as DoS attack'* [10].

Considering this fact, once the network traffic is clustered by *x-means*, and then following the assumption 3 of clustering based anomaly detection, we give priority to the smaller clusters for being anomalous. That means, we sort the produced clusters according to their size. Next, in each cluster, we separate the payload length attribute and apply *x-means* clustering again. Then we calculate the variance of each cluster produced and the corresponding records of the cluster with minimum variance are identified as potential DoS attack. Basically, the group of corresponding records in the cluster with identical <*SrcIP, DstIP, Protocol, Payload Length*> are considered as collective anomaly. In this way, *x-means* is applied on all the original clusters produced earlier and collective anomaly is detected.

TABLE II
NETWORK TRAFFIC SAMPLE

SrcIP	*DstIP*	*Protocol*	*Payload Length*
207.75.239.115	172.16.114.50	TCP	1460
207.75.239.115	172.16.114.50	TCP	1460
207.75.239.115	172.16.114.50	TCP	1460
207.75.239.115	172.16.114.50	TCP	1460
207.75.239.115	172.16.114.50	TCP	1460
207.75.239.115	172.16.114.50	TCP	1460
207.75.239.115	172.16.114.50	TCP	1460
135.13.216.191	172.16.113.105	TCP	22
135.13.216.191	172.16.113.105	TCP	22
135.13.216.191	172.16.113.105	TCP	37
135.13.216.191	172.16.113.105	TCP	38
135.13.216.191	172.16.113.105	TCP	39
135.13.216.191	172.16.113.105	TCP	40
135.13.216.191	172.16.113.105	TCP	41
135.13.216.191	172.16.113.105	TCP	22

Let us give an example to illustrate the CAD algorithm. In Table 2, a sample cluster of network traffic flow is displayed after performing *x-means* algorithm. In the next step, *x-means* algorithm is applied again on the the payload length attribute instances to find the clusters within and clustering results yield two clusters c_1, c_2, where c_1=(1460,1460,1460,1460,1460,1460) and c_2=(22,22,37,38,39,40,41,22). Next, we calculate the variance (8) of each cluster and consider the corresponding records of cluster with minimum variance as attack traffic. Table 3 displays that variance of c_1 is minimum and thus the corresponding flow of the cluster c_1 are considered as attack such as *<207.75.239.115,172.16.114.50,TCP,1460>* is an attack traffic flow.

$$Var(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2 \quad (8)$$

where, x_i is an instances and $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$, which is the mean of all the instances.

TABLE III
VARIANCE OF CLUSTERS c_1,c_2

Cluster	*Variance*
c_1:(1460,1460,1460,1460,1460,1460)	0
c_2:(22,22,37,38,39,40,41,22)	78.83

VI. EXPERIMENTAL ANALYSIS

In this section we validate our algorithm with benchmark 1998 DARPA dataset from MIT Lincoln Lab [11]. We used Rapid Miner tool [12] for implementing *x-means* algorithm in MAC OS Mavericks with core i7 processor and 8GB DDR3 RAM. The experimental section is divided into two parts. In the first part we give a brief description about the dataset used and the preprocessing steps. In the second part, we compare our approach against existing techniques. Then, we give a brief idea about run time complexity.

A. From Ocean to Pond

In order to examine the ability of our approach to identify interesting patterns, we require network data containing known traffic patterns. However, traffic data with known patterns are not widely available. We decided to use a widely accepted synthetic dataset for intrusion detection from MIT Lincoln Lab. The 1998 DARPA dataset provides traffic files of 7 weeks that contain raw traffic flow records with an associated label to indicate whether the records was part of a normal or attack flow. In our experiment, we used traffic files from 7th week's Friday. The rationale behind using these files is the presence of DoS attack and volume. Overwhelming majority of top ten attacks in DARPA dataset are caused by Denial of Service attack [13]. Particularly, the dataset we used contains *Neptune, Smurf* and *Back* attacks, which are all DoS category.

Due to the growth in bandwidth of networks, the volume of network traffic data is often too large for analysis using traditional data mining techniques [14]. Sampling is a popular method for improving the scalability of analyzing massive dataset such as network traffic. For our experiment, we took a subset of week 7, Friday DARPA traffic containing top DoS attacks according to frequency. However, our future work will include the investigation of summarization techniques to be applied on network traffic. We used ipsumdump tool [15] to extract the raw data and converted to human readable format. For clustering, we converted the IP addresses into integer since the integers can uniquely represent them and the protocols are replaced with numbers, i.e., *TCP=1, UDP=2, ICMP=3*.

B. Accuracy

We measure the accuracy of our approach using the standard confusion metrics. The metrics are listed as True Positive (TP), False Positive (FP), True Negative (TN), False negative (FN). Table 4 displays the confusion metrics.
TP = Attack correctly identified as attack.
FP = Normal traffic incorrectly identified as attack.
TN = Normal traffic correctly identified as normal.
FN = Attack incorrectly identified as normal.

TABLE IV
STANDARD CONFUSION METRICS FOR EVALUATION OF CAD ALGORITHM

Actual traffic label	**Normal**	**Attack**
Normal	TN	FP
Attack	FN	TP

The accuracy is computed using equation (9)

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

The results were compared against CBLOF [7], LDCOF [8] and *k-means* algorithm, where the clustering based anomaly detection assumptions are used. Figure 4 shows, using experimental results that the proposed approach CAD (97%) performs significantly better than both *k-means* (85%), CBLOF (83%) and LDCOF (83%).

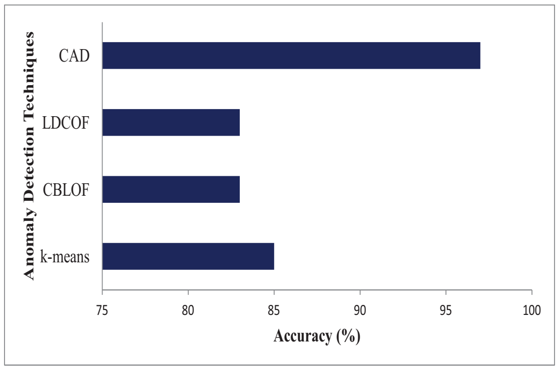

Fig. 4. Accuracy comparison

Table 5 also displays the ability of individual techniques to detect the present attacks in the dataset used. Our approach could detect all three types of attack present, whereas *k-means* detected two types (*Neptune, Smurf*), CBLOF and LDCOF detected only one type (*Back*) of attack.

TABLE V
ATTACK DETECTION COMPARISON AMONG INDIVIDUAL TECHNIQUES

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

C. Run Time Complexity

The rationale behind choosing *x-means* algorithm also includes run time complexity. The *x-means* algorithm was compared with highly optimized but traditional implementations of *k-means* algorithm with different values of *k*. In an experiment of Las Campanas Redshift Survey [16], it is found that *x-means* is eight times faster than *k-means* algorithm. However, we also compared the run time of both algorithms and the result is shown in Figure 5.

VII. Conclusion

In this paper, we introduced the concept of collective anomaly for network traffic analysis. We used a variant of *k-means* algorithm, *x-means* algorithm, for clustering network traffic and detect DoS attacks. The key contribution of our scheme is the incorporation of collective anomaly for detecting Denial of Service attack, which is a novel approach. Experimental results demonstrate that it outperforms existing techniques in detection accuracy on benchmark DARPA dataset. In future, we will develop an efficient sampling technique which

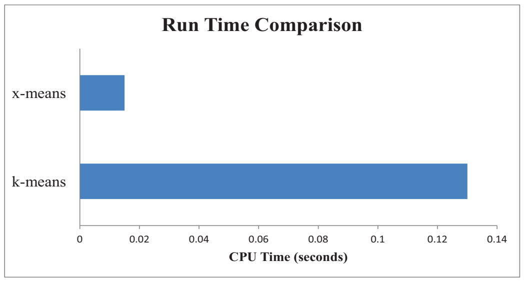

Fig. 5. Runtime comparison between *k-means* and *x-means*

will capture the underlying distribution of the data and create a summary to be able to monitor high capacity networks. We will also extend our approach to support online data streams for efficiently identifying interesting patterns.

References

[1] K. Kendall, "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)*, 1999, pp. 12–26.

[2] "1999 kdd cup dataset," accessed: 2013-12-21. [Online]. Available: www.kdd.ics.uci.edu

[3] A. M. Dan Pelleg, "X-means: Extending k-means with efficient estimation of the number of clusters," in *Proceedings of the Seventeenth International Conference on Machine Learning*. San Francisco: Morgan Kaufmann, 2000, pp. 727–734.

[4] M. Ahmed and A. Naser, "A novel approach for outlier detection and clustering improvement," in *Industrial Electronics and Applications (ICIEA), 2013 8th IEEE Conference on*, 2013, pp. 577–582.

[5] V. Hautamäki, S. Cherednichenko, I. Kärkkäinen, T. Kinnunen, and P. Fränti, "Improving k-means by Outlier Removal," in *Proc. 14th Scandinavian Conference on Image Analysis (SCIA'05)*, 2005, pp. 978–987.

[6] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, L. M. L. Cam and J. Neyman, Eds., vol. 1. University of California Press, 1967, pp. 281–297.

[7] Z. He, X. Xu, and S. Deng, "Discovering cluster based local outliers," *Pattern Recognition Letters*, vol. 2003, pp. 9–10, 2003.

[8] M. G. Mennatallah Amer, "Nearest-neighbor and clustering based anomaly detection algorithms for rapidminer." Shaker Verlag GmbH, Aachen, 8 2012, pp. 1–12.

[9] K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate Analysis*. Academic Press, 1979.

[10] K. Wang and S. J. Stolfo, "Anomalous payload-based network intrusion detection." in *RAID*, ser. Lecture Notes in Computer Science, E. Jonsson, A. Valdes, and M. Almgren, Eds., vol. 3224. Springer, 2004, pp. 203–222.

[11] "1998 darpa dataset," accessed: 2013-12-21. [Online]. Available: www.ll.mit.edu

[12] "RapidMiner," Online, Apr. 2012. [Online]. Available: http://rapid-i.com/content/view/181/190/

[13] A. Mahmood, C. Leckie, and P. Udaya, "A scalable sampling scheme for clustering in network traffic analysis," in *Proceedings of the 2nd International Conference on Scalable Information Systems*, ser. InfoScale '07, 2007, pp. 38:1–38:8.

[14] K. C. Claffy, G. C. Polyzos, and H.-W. Braun, "Application of sampling methodologies to network traffic characterization," *SIGCOMM Comput. Commun. Rev.*, vol. 23, no. 4, pp. 194–203, Oct. 1993.

[15] "Ipsumdump tool," accessed: 2013-12-21. [Online]. Available: www.cs.ucla.edu/ kohler/ipsumdump

[16] "Las campanas redshift survey," accessed: 2013-12-21. [Online]. Available: http://www.manaslu.astro.utoronto.ca/ lin