A Study of K-Means and C-Means Clustering Algorithms for Intrusion Detection Product Development

Santosh Kumar Sahu and Sanjay Kumar Jena

***Abstract*—The increase in Internet and Internet based application, the business premises have now spread throughout the world. Due to the extreme competitions among the business, one tries to demolish other. Hence, secure product design techniques should be adopted. To protect the applications from intruder, intrusion detection system becomes utmost requirement for every organization. In intrusion detection models enormous quantity of training data is required. As a result, sophisticated algorithms and high computational resources are required. In Intrusion Detection System, to separate normal activities from abnormal activities clustering algorithms are used. To select an efficient clustering algorithm is a challenging task. In this paper, a comparison has been made between K-Means and C-Means clustering on intrusion datasets. The simulation contains all proximity measures of K-Means and C-Means clustering techniques. The accuracy of these clustering algorithms is compared using the confusion matrix. The result shows that K-Means provides better clustering accuracy in comparison with C-Means. Therefore, to design intelligent intrusion detection product K-Means is a better option.**

***Index Terms*—K-Means, C-Means, KDD Cup99, GureKDD, NSLKDD.**

I. Introduction

In the present day it is highly essential to design intelligence software products which can withstand zero day attacks. The innovative product development is utmost essential to every software firm. They should focus on how the product is survive in an insecure medium like the internet. Interdisciplinary concepts are required to tolerate the unusual activities.

The term intrusion comprises a set of attempts to compromise the confidentiality, integrity and availability of information resources. Intrusion detection is the process of monitoring the events in the system and analyzing the network packets to or from the network. Intrusion detection system automates the process and counteract the intrusive efforts. The intrusive efforts can be caused by insiders or outsiders in the system. The intruder can be classified as clandestine, misfeasor and masquerader [1]. The advance of internet technology makes life easier in the field of communication and interaction between human and computer. However the attacker tries to find the vulnerability in the internet based application and try to penetrate it. The prime duty of the information security research community is to monitor, detect and prevent the intrusive efforts.

The Intrusion Detection System (IDS) techniques can be broadly categorized into two types on the basis of the detection methodologies viz. signature based and anomaly based. The signature based IDS detect only the known attacks whose signatures are stored in the database. The anomaly based IDS compares the definition of activities which are considered as normal as against the observed events to identify signature deviation. The anomaly based IDS generates many false alarms, which degrades the performance. The traditional IDS is rule based. The implementer writes rules for normal and abnormal conditions. As per the rule condition the IDS detect the intrusions. It is good enough to find all known attacks whose rules are exist in their database. However, it is not efficient to detect unknown attacks and the existing changed attack patterns.

It becomes utmost essential to design an IDS that can detect known and unknown attacks. By combining both techniques, we can design a hybrid detection approach that improves the intrusion detection process. But, it requires a large amount of data for training and testing. To design a hybrid intrusion detection system, classification and clustering techniques are used who can classify the normal and unusual packets present in the network traffic. To design an IDS which can detect the unknown attacks, researchers used data mining and machine learning algorithms. Unsupervised classification algorithms are used to separate normal and abnormal activities exist in the network traffic. The widely used unsupervised classifier are K-Means and Fuzzy C-Means.

In this paper, K-Mean and C-Mean clustering techniques are implemented and tested on three intrusion datasets namely KDDCup99 [2], NSLKDD [3], and GureKDD [4]. The datasets are preprocessed and normalized using various data preprocessing techniques [1], and then applied as input to the models. K-Means and C-Means clustering algorithms are analyzed based on their clustering accuracy and computational time.

The remaining of this paper is organized as follows. Section II gives a brief idea about K-Means and C-Means clustering techniques, Section III presents the implementation of different dissimilarity measures using K-Means and C-Means. Section IV presents the comparative results and finally, concluding remarks are given in Section V.

II. Clustering Techniques

Cluster analysis categorizes the data object based on the information that describes the objects and their relationships.

Manuscript received April 4, 2014; revised May 30, 2014.

The authors are with the National Institute of Technology, Rourkela, Odisha, 769008 India (e-mail: santoshsahu@hotmail.co.in, skjena@nitrkl.ac.in).

DOI: 10.7763/IJIMT.2014.V5.515

The primary goal of clustering is to separate the similar and dissimilar objects. The efficiency of a clustering algorithm depends the similarity measure of the objects belonging to a cluster.

A. Types of Clusters

We can categorize the clustering techniques into the following types

1) Well Separated: The objects present in the dataset is grouped without overlapping and the clusters are separated from each other by a distance.
2) Prototype based: A cluster is a set of objects in which each object within a cluster closer to the prototype that defines the cluster than to the prototype of any other cluster. The prototype may be centroid or mean in case of continuous data and centroid in case of categorical data.
3) Graph Based: The data are represented as a graph, where the nodes are treated as objects and the edges are represented as connections among objects. A cluster can be defined as a connected component. It means that the objects are connected within a cluster and have no connection from outside cluster.
4) Density Based: A cluster is a dense region of objects that is surrounded by a region of low density. It is applicable for irregular, inter wined, noisiest and outliers present in the dataset.
5) Shared-Property clusters: We can construct a cluster as a set of objects that share common properties among them. The property may be statistically or mathematically related among the objects.

In this paper we have considered two prototypes based clustering algorithm, namely K-Means and C-Means.

1) K-Means

TABLE I: NUMBER OF SAMPLES BEFORE AND AFTER DATA PREPROCESSING

Dataset	No of Samples before data preprocessing.	Samples after data preprocessing.	% reduction
KDD Full	4898431	1074992	78.05%
KDD 10%	494021	145586	70.53%
KDD Corrected	311029	77291	75.15
NSLKDD Train	25192	25192	0%
NSLKDD Test	22544	22544	0%
GureKDD 6% dataset	178835	160904	10.03%

K-Means is a partitioning based clustering method which analyzes data and treats the data objects based on locations and distances between various input data points. It creates K number of clusters from N number of observations where K is less than or equal to N. The K-Means algorithm is given in Fig. 1. As per the objective function it finds the dissimilarity or distance between the data objects and predict their cluster of an object which has minimum distance [5]. Different kind of dissimilarity measures is represented in [8], [9].

K-Means compute centroid clusters differently for the different supported distance measures. As per the proximity measures the objective function is calculated. We have implemented four distance measures, namely: L1, L2, cosine and correlation. L1 is known as Manhattan distance. Each centroid is the component-wise median of the points in that cluster [7]. L2 is the squared Euclidean distance. Each centroid is the mean of the points in that cluster [6].

Algorithm K-Means

```
Kmeans (dataset, K, dissimilarity_measure, replicate)
Step 1: randomly select a K number of initial centroids.
Step 2: repeat
        Construct K clusters, as per the dissimilarity
        measure/objective function
        Re-compute the centroid of each cluster
    Until the number of replicates given/centroids do
    not change
Step 3  End
```

Fig. 1. K-Means algorithm.

In Cosine proximity measure the distance is measured as one minus the cosine of the included angle between points (known as vectors). The centroid is the mean of the points in that cluster, after normalizing those points to the unit Euclidean length. The correlation dissimilarity measure is calculated as one minus the sample correlation between points (treated as a sequence of values). To find the centroid of a cluster, first centering and then normalizing the points to zero mean and unit standard deviation. Each centroid of a cluster is the component-wise mean of the data points.

2) C-Means

The Fuzzy C-means (FCM) clustering algorithm is one of the most popular fuzzy clustering techniques, which was originally proposed by Dunn *et al.* [10] and later had been modified by Bezdek *et al.* [11]. The Fuzzy C-Means algorithm is given in Fig. 2. Fuzzy C-Means are able to determine, and update the membership values iteratively of the data points with pre-defined number of cluster i.e. K. Thus, every data point present in the dataset carries a membership value for all clusters. FCM has been extensively used in various fields which is discussed in [12]-[15]. A large number of variants of the FCM algorithm had been proposed. Sikka *et al*. [16] discussed some of these algorithms. In this paper, we have implemented three basic options of FCM and compared their accuracy with K-Means algorithm.

III. IMPLEMENTATION

To test the efficiency of K-Means and C-Means, the well-known benchmarked intrusion datasets are used. These datasets are used by the researchers of information security for the empirical analysis of intrusions in network security. Before being fed to the dataset it must be properly processed. Data preprocessing is one of the most important and time consuming process. The data may be captured from various repositories. It becomes utmost essential to convert the data into an appropriate format before passing through the algorithm [1]. The various steps for preprocessing the data includes filling the missing values, removing redundant records, balancing the dataset, selecting most relevant feature and normalize the instances. Min-max normalization applied to represent the data elements within 0 to 1. Table I contains the number of samples, before and after data preprocessing technique has applied.

Algorithm C-Means

C-Means (dataset, K, Options)

Step 1: Fix c (2<=c<=n) and select a value for parameter m. Initialize the partition matrix $U^{(0)}$.Each step in this algorithm will be labelled r, where r = 0,1,2.......

Step 2: Calculate the center c { V_{ij} } for each step.

$$V_{ij} = \frac{\sum_{k=1}^{n} (\mu_{ik})^m x_{kj}}{\sum_{k=1}^{n} (\mu_{ik})^m}$$

Step 3: Calculate the distance matrix $D_{[c,\, n]}$

$$D_{ij} = \left[\sum_{j=1}^{m} (\mathrm{x}_{kj} - \mathrm{v}_{ij})^2 \right]^{\frac{1}{2}}$$

Step 4: Update the partition matrix for the r^{th} step, $U^{(r)}$ as follows

$$\mu_{ij}^{r-1} = \left[\frac{1}{\sum_{j=1}^{c} (\mathrm{d}_{ik}^{r} / \mathrm{d}_{jk}^{r})^{\frac{2}{m-1}}} \right]$$

If $\|U^{(k+1)} - U^{k}\| < \delta$ then "Stop" otherwise return to step 2 by iteratively updating the cluster centers and the membership grades for data point.

Fig. 2. C-Means algorithm.

The original KDDCup Full dataset contains 4898431 numbers of samples. After applying data preprocessing the number of samples reduced to 1074992 and the percentage of reduction is 78.05. Table I shows the percentage of reduction for KDD, NSLKDD and GureKDD dataset. The processed datasets applied to the clustering algorithms. The detail implementation of the clustering algorithms with different dissimilarity measures are given below:

A. K-Means

There are four dissimilarity measures are implemented using K-Means. Three intrusion datasets used as input. To measure the accuracy of different objective functions, we have calculated the confusion matrix for each proximity measure using these datasets which is given in Fig. 3-Fig. 5.

B. C-Means

In C-Means clustering technique, the following four options are present. The details about the options are as follows [18]:

OPTIONS (1): exponent for the given dataset matrix. The default value is 2.0.

OPTIONS (2): maximum number of iterations to form the clusters (default number of iterations is 100).

OPTIONS (3): minimum amount of improvement in execution (default value is 1×10^{-5}).

OPTIONS (4): info display during iteration for Fuzzy C-Mean (default value is 1).

In our experiment, we have considered option 2, 3 and 4 to construct the confusion matrix on intrusion datasets in order to measure the performance of C-Means algorithm. The option 1 is skipped because it only works with binary data. The following datasets are used in our experiment.

1) KDD corrected dataset

The KDD corrected refined dataset contains 77291 number of samples as per Table 1. It was supplied as input to the four objective functions of K-Means and obtains the confusion matrix. The efficiency, recall, sensitivity, specificity and negative predicted value are calculated using confusion matrix. The details of confusion matrix are discussed in [1].

2) NSLKDD dataset

In NSLKDD dataset is available in three forms as full dataset, train 20% and test dataset. The dataset does not contain any duplicate records. In data preprocessing, we applied min-max normalization technique to normalize the dataset. For analysis, we have considered two datasets i.e. train and test dataset as given in Table I. These datasets are input to K-Means to obtain the confusion matrix.

3) GureKDD

The size of GureKDD dataset is very large that is up to 9GB. Therefore, we have considered the 6% dataset and applied data preprocessing. As per the Table I GureKDD 6% dataset contains 160904 numbers of samples. This dataset is input to K-Means to obtain the confusion matrix.

The accuracy, computational time, best total of distance using different proximity measures is given in Table II. The Table III contains variety options of C-Means, computational time, distance of the objective function and clustering

TABLE II: IMPLEMENTATION OF DIFFERENT PROXIMITY MEASURES K-MEANS WITH TIME SUM OF THE DISTANCE AND CLUSTERING ACCURACY

Sl. No.	Dataset	Dissimilarity Measure of K-Means	Time in Sec.	Best total sum of distance	Accuracy in %.
1	KDD corrected	L1	4.56	148337	90.5
		L2	2.99	85900	91.4
		Cosine	2.09	8176.31	91.2
		Correlation	2.10	10189.7	91.1
2	NSLKDD Train	L1	1.55	68216.2	81.0
		L2	0.99	39599.2	88.48
		Cosine	0.76	3298.61	88.5
		Correlation	0.74	4240.59	88.5
3	GureKDD	L1	10.1	377213	82.6
		L2	7.99	206036	76.5
		Cosine	10.1	25060	77.3
		Correlation	6.8	30062	77.4

IV. Result and Discussion

The K-Means and C-Means clustering algorithms has been investigated using various proximity measures on intrusion datasets. It is useful to summarize the results and presented the comparison of their performances.

To compare results of K-Means and C-Means, first we select whose dissimilarity measure which provides better result. For example, The Euclidean distance measure provides more accuracy in comparison with others using KDD Corrected dataset as given in Fig. 3. Similarly, the option 2 using K-Means provides most favorable results as given in Fig. 4. The comparative analysis of K-Means and C-Means clustering using KDD Corrected dataset, we select Euclidean distance measure for K-Means and option 2 using C-Means and depicted result in Fig. 9. The Fig. 10-Fig. 12 contained the comparative result using NSLKDD and GureKDD intrusion dataset.

TABLE III: Implementation of Different Options of C-Means with Time, Distance of the Objective Function and Clustering Accuracy

SL no	Dataset	Options of C-Means	Time	Objective Function	Accuracy
1	KDD corrected	2	2.48	62077.57	91.0
		3	3.43	35047.63	90.9
		4	4.8	18648.40	91.0
2	NSLKDD Train	2	0.08	27112.40	87.7
		3	1.16	14929.90	87.9
		4	1.44	7807.43	88.1
3	GureKDD	2	8.15	133400.1	58.6
		3	9.68	69274.27	55.8
		4	16.1	35054.62	55.3

A. KDD Corrected Dataset

The Euclidean distance provides better accuracy in comparison with the other proximity measures on KDD Corrected dataset as given in Fig. 1. The Fig. 1 contains four confusion matrixes for four proximity measures used in K-Means algorithm. The accuracy of K-Means algorithm is more desirable using Euclidean distance among the data points as given in Fig. 3 (d). The Fig. 6 contains a confusion matrix for C-Means based on three options. The accuracy of option 2 and 4 are equal. For evaluation of the results, the Euclidean distance measure for K-Means and C-Means with option 2 has been considered. In Fig. 9 shows the accuracy of K-Means and C-Means by taking the number of samples in X axis and accuracy in Y axis. We have divided the dataset into ten parts and applied these two algorithms and drawn the accuracy in Fig. 9. The accuracy of K-Means are slightly better in comparison with C-Means using KDD Corrected dataset.

B. NSLKDD Dataset

It has been observed that the correlation and the cosine dissimilarity measure provide better results in comparison with the L1 and L2 distance measure on K-Means as given in Fig. 5. C-Means with option 4 provides favorable results among all options as given in Fig. 6.

To find the best option between K-Means and C-Means, we have considered correlation measures for K-Means and option 4 for C-Means and drawn the result in Fig. 10. The overall efficiency of K-Means is better in comparison with C-Means for all sets of data points as given in Fig. 10.

47726 61.7%	**6719** 8.7%	**87.7%** 12.3%
187 0.2%	**22659** 29.3%	**99.2%** 0.8%
99.6% 0.4%	**77.1%** 22.9%	**91.1%** 8.9%

(a)

47707 61.7%	**6560** 8.5%	**87.9%** 12.1%
206 0.3%	**22818** 29.5%	**99.1%** 0.9%
99.6% 0.4%	**77.7%** 22.3%	**91.2%** 8.8%

(b)

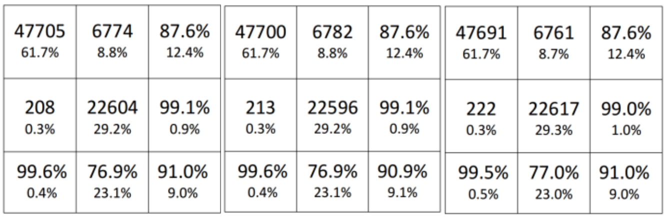
47781 61.8%	**7201** 9.3%	**86.9%** 13.1%
132 0.2%	**22177** 28.7%	**99.4%** 0.6%
99.7% 0.3%	**75.5%** 24.5%	**90.5%** 9.5%

(c)

47675 61.7%	**6411** 8.3%	**88.1%** 11.9%
238 0.3%	**22967** 29.7%	**99.0%** 1.0%
99.5% 0.5%	**78.2%** 21.8%	**91.4%** 8.6%

(d)

Fig. 3. Confusion matrix of K-Means clustering: a) correlation, b) cosine c) manhattan d) euclidean distance on KDD corrected dataset.



47705 61.7%	6774 8.8%	87.6% 12.4%
208 0.3%	22604 29.2%	99.1% 0.9%
99.6% 0.4%	76.9% 23.1%	91.0% 9.0%

(a)

47700 61.7%	6782 8.8%	87.6% 12.4%
213 0.3%	22596 29.2%	99.1% 0.9%
99.6% 0.4%	76.9% 23.1%	90.9% 9.1%

(b)

47691 61.7%	6761 8.7%	87.6% 12.4%
222 0.3%	22617 29.3%	99.0% 1.0%
99.5% 0.5%	77.0% 23.0%	91.0% 9.0%

(c)

Fig. 4. Confusion matrix of the fuzzy C-Means clustering a) option 2, b) option 3 and c) option 4 on KDD corrected dataset.

13402 53.2%	**2852** 11.3%	**82.5%** 17.5%
47% 0.2%	**8891** 35.3%	**99.5%** 0.5%
99.7% 0.3%	**75.7%** 24.3%	**88.5%** 11.5%

(a)

13397 53.2%	**2849** 11.3%	**82.5%** 17.5%
52% 0.2%	**8894** 35.3%	**99.4%** 0.6%
99.6% 0.4%	**75.7%** 24.3%	**88.5%** 11.5%

(b)

13402 53.2%	**2890** 11.5%	**82.3%** 17.7%
47 0.2%	**8853** 35.1%	**99.5%** 0.5%
99.7% 0.3%	**75.4%** 24.6%	**88.3%** 11.7%

(c)

13421 53.3%	**4753** 18.9%	**73.8%** 26.2%
28 0.1%	**6990** 27.7%	**99.6%** 0.4%
99.8% 0.2%	**59.5%** 40.5%	**81.0%** 19.0%

(d)

Fig. 5. Confusion matrix of K-Means clustering a) correlation, b) cosine c) manhattan d) euclidean distance on the NSLKDD train dataset.

13401 53.2%	**3039** 12.1%	**81.5%** 18.5%
48 0.2%	**8704** 34.6%	**99.5%** 0.5%
99.6% 0.4%	**74.1%** 25.9%	**87.7%** 12.3%

(a)

13392 53.2%	**2980** 11.8%	**81.8%** 18.2%
57 0.2%	**8763** 34.8%	**99.4%** 0.6%
99.6% 0.4%	**74.6%** 25.4%	**87.9%** 12.1%

(b)

13388 53.1%	**2938** 11.7%	**82.0%** 18.0%
61 0.2%	**8805** 35.0%	**99.3%** 0.7%
99.5% 0.5%	**75.0%** 25.0%	**88.1%** 11.9%

(c)

Fig. 6. Confusion matrix of the fuzzy C-Means clustering a) option 2, b) option 3 and c) option 4 on NSLKDD dataset.

The Fig. 11 contains the comparative accuracy between K-Means and C-Means using NSLKDD test dataset. The points showed that K-Means provide better result in comparison with C-Means algorithm using NSLKDD Test dataset.

C. *GureKDD Dataset*

122118 75.9%	**1395** 0.9%	**98.9%** 1.1%
34930 21.7%	**2461** 1.5%	**93.4%** 6.6%
77.8% 22.2%	**63.8%** 36.2%	**77.4%** 22.6%

(a)

121834 75.7%	**1381** 0.9%	**98.9%** 1.1%
35214 21.9%	**2475** 1.5%	**93.4%** 6.6%
77.6% 22.4%	**64.2%** 35.8%	**77.3%** 22.7%

(b)

131844 81.9%	**2766** 1.7%	**97.9%** 2.1%
25204 15.7%	**1090** 0.7%	**95.9%** 4.1%
84.0% 16.0%	**71.7%** 28.3%	**82.6%** 17.4%

(c)

120669 75.0%	**1379** 0.9%	**98.9%** 1.1%
36379 22.6%	**2477** 1.5%	**93.6%** 6.4%
76.8% 23.2%	**64.2%** 35.8%	**76.5%** 23.5%

(d)

Fig. 7. Confusion matrix of K-Means clustering a) correlation, b) cosine c) manhattan d) euclidean distance on GureKDD dataset.

The Manhattan distance measure of K-Means provides better accuracy in comparison to other measures as given Fig. 7(c). The accuracy of C-Means with option 2 is more than the other options as depicted in Fig. 8(a). For selecting the best alternatives of K-Means and C-Means we have considered the Euclidean distance for K-Means and option 2 with C-Means. The comparison result is drawn in Fig. 12. Experimentally, it is clearly showing that the accuracy of C-Means is very less in comparison with K-Means. For example, maximum accuracy of C-Means and K-Means are 60.5% and 83.9% respectively.

90751 56.4%	**263** 0.2%	**99.7%** 0.3%
66297 41.2%	**3593** 2.2%	**94.9%** 5.1%
57.8% 42.2%	**93.2%** 6.8%	**58.6%** 41.4%

(a)

86251 53.6%	**254** 0.2%	**99.7%** 0.3%
70797 44.0%	**3602** 2.2%	**95.2%** 4.8%
54.9% 45.1%	**93.4%** 6.6%	**55.8%** 44.2%

(b)

85298 53.0%	**254** 0.2%	**99.7%** 0.3%
71750 44.6%	**3602** 2.2%	**95.2%** 4.8%
54.3% 45.7%	**93.4%** 6.6%	**55.3%** 44.7%

(c)

Fig. 8. Confusion matrix of fuzzy C-Means clustering a) option 2, b) option 3 and c) option 4 on GureKDD dataset.

As per the Fig. 12 K-Means provide more favorable result in GureKDD dataset. This experiment reveals the fact that the accuracy of these algorithms depends on the distribution of the data points. To achieve high accuracy the data distribution should be consistent in all manners. Again the algorithm should be chosen as per the problem in hand.

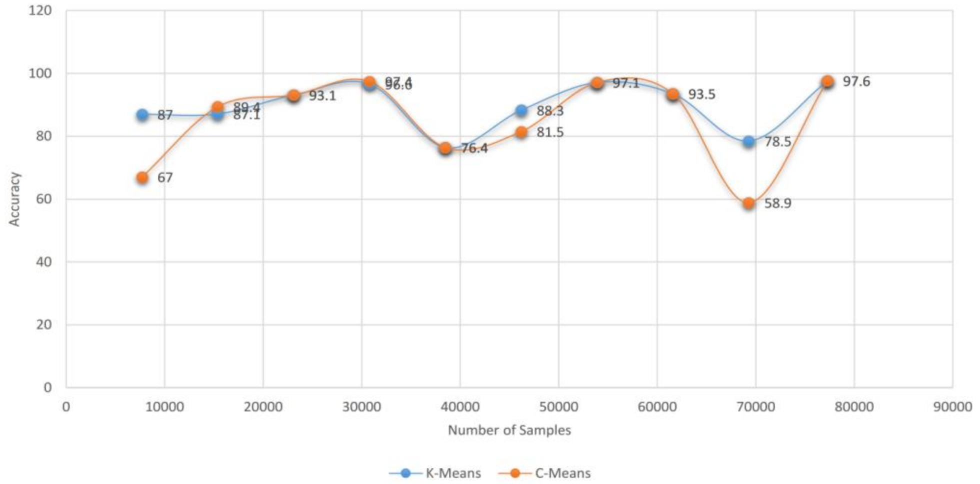


Fig. 9. Comparison of K-Means and C-Means on KDD corrected dataset.

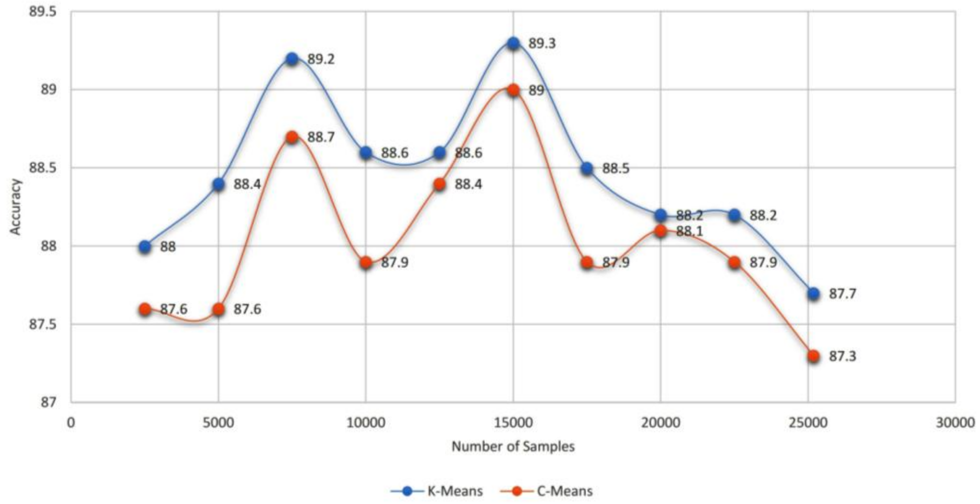


Fig. 10. Comparison of K-Means and C-Means on NSLKDD train dataset.

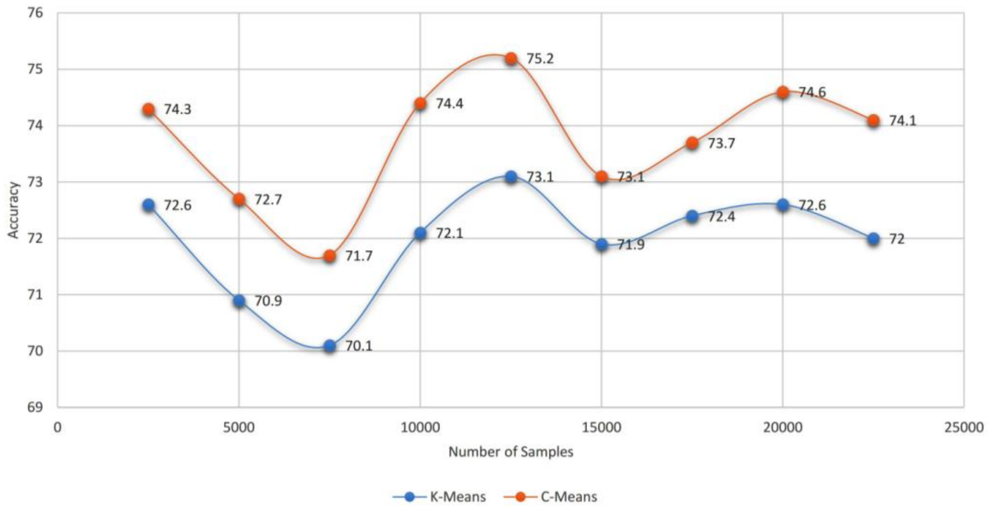


Fig. 11. Comparison of K-Means and C-Means on NSLKDD test dataset.

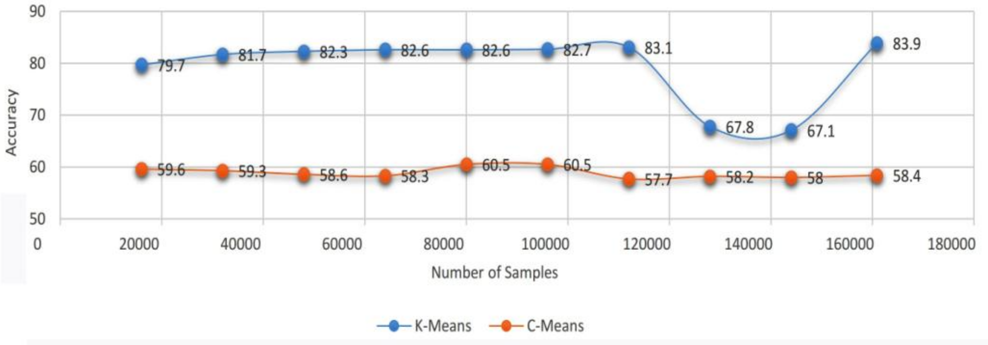


Fig. 12. Comparison of K-Means and C-Means on GureKDD dataset.

V. Conclusion

Two clustering techniques based on intrusion datasets have been reviewed in this paper. These clustering techniques with different similarity measures are implemented, evaluated and compared using intrusion datasets. The comparative study discussed here is concerned with the accuracy of each algorithm, with care being taken towards the accuracy in calculation and other performance related measures. It is found that the K-Means clustering algorithm provides better accuracy and consumes less time in comparison to C-Means clustering on these datasets.

The clustering techniques discussed here don't have to be used alone to predict different attacks. As the initial centroids are chosen randomly, the class distribution may change or evolve on each execution. Therefore, it should be used in conjunction with other data mining algorithms for better accuracy.

References

[1] Sahu, S. Kumar, S. Sarangi, and S. K. Jena. A detail analysis on intrusion detection datasets. Advance Computing Conference (IACC), 2014 IEEE International, 2014.

[2] KDD Cup 1999 Dataset. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/

[3] NSL-KDD dataset. [Online]. Available: http://nsl.cs.unb.ca/NSL-KDD/

[4] GureKDDCup Dataset. [Online]. Available: http://www.sc.ehu.es/acwaldap/

[5] Jang *et al.*, *Neuro-Fuzzy and Soft Computing- A computational Approach to Learning and Machine Intelligence*, Prentice Hall.

[6] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, New York: John Wiley and Sons, 1973.

[7] H. Kashima, J. Hu, B. Ray, and M. Singh, "K-means clustering of proportional data using L1 distance," in *Proc. 19th International Conference on Pattern Recognition*, pp.1-4, 8-11 Dec. 2008.

[8] I. S. Dhillon, S. Mallela, and R. Kumar, "A divisive information-theoretic feature clustering algorithm for text classification," *Journal of Machine Learning Research*, vol. 3, pp. 1265–1287, 2003.

[9] I. S. Dhillon and D. S. Modha, "Concept decompositions for large sparse text data using clustering," *Machine Learning*, vol. 42, pp. 143–175, 2001.

[10] J. C. Dunn, "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters," *J. Cybernet*, vol. 3, 1973, pp. 32–57.

[11] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Norwell, MA, USA: Kluwer Academic Publishers, 1981.

[12] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy C-Means model," *IEE EFS*, vol. 3, no. 3, p. 370, 1995.

[13] D. Q. Zhang and S. C. Chen, "A novel kernelized fuzzy C-Means algorithm with application in medical image segmentation," *Artif. Intel. Med*, vol. 32, pp. 37–50, 2004.

[14] S. Migaly, J. Abonyi, and F. Szeifert, "Fuzzy Self-Organizing Map Based on Regularized Fuzzy C-Means Clustering," in *Advances in Soft Computing, Engineering Design and Manufacturing*, J. M. Benitez, O. Cordon, F. Hoffmann *et al.* (Eds.), Springer Engineering Series, (Revised papers of the 7th On-line World Conference on Soft Computing in Industrial Applications (WSC7)), 2002, pp. 99–108.

[15] S.Albayrak and F. Armasyali, "Fuzzy C-Means Clustering on Medical Diagnostic System," in *Proc. Int. XII Turkish Symp, Artif. Intel. NN*, 2003.

[16] K. Sikka, N. Sinha, P. K. Singh, and A. K. Mishra, "A Fully Automated Algorithm under Modified FCM Framework for Improved Brain MR Image Segmentation," *Magnetic Resonance Imaging*, vol. 27, no. 7, pp. 994–1004, 2009.

[17] S. Chattopadhyay, D. K. Pratihar, and S. C. De Sarkar, "A Comparative Study of Fuzzy C-Means Algorithm and Entropy-Based Fuzzy Clustering Algorithms," *Computing & Informatics*, vol. 30, no. 4, 2011.

[18] Matlab. [Online]. Available: http://www.mathworks.in

Santosh Kumar Sahu received the B.Tech degree from C.V. Raman College of Engineering, Bhubaneswar, M.Tech from Berhampur University. Since 2012, he has been a Research Scholar (Ph.D.) with National Institute of Technology, Rourkela, Odisha, India. His research interests include intrusion detection system, digital forensic and cyber security.

Sanjay Kumar Jena is currently a professor in the Department of Computer Science and Engineering at National Institute of Technology, Rourkela, Odisha, India. He received his M.Tech and Ph.D from IIT Kharagpur and IIT Bombay, respectively. His research interests include information security, network security, data privacy and distributed systems. He has published over 90 research papers in various journals and conferences of repute.