An analysis of suitable parameters for efficiently applying K-means clustering to large TCPdump data set using Hadoop framework

Jakrarin Therdphapiyanak
Dept. of Computer Engineering
Chulalongkorn University
Bangkok, Thailand
Email: Jakrarin.Th@student.chula.ac.th

Krerk Piromsopa
Dept. of Computer Engineering
Chulalongkorn University
Bangkok, Thailand
Email: krerk.p@chula.ac.th

***Abstract*—In this paper, we determined the appropriate number of clusters and the proper amount of entries for applying K-means clustering to TCPdump data set using Apache Mahout/Hadoop framework. We aim at finding suitable configuration for efficiently analyzing large data set in limited amount of time. Our implementation applied Hadoop for large-scale log analysis with data set from KDD'99 competition as test data. With the distributed system framework, we can analyze a whole data set of KDD'99 by first applying our preprocessing. In addition, we use an anomaly detection model for log analysis. A key challenge is to make anomaly detection work more accurately. For the K-means algorithm, a key challenge is to set the appropriate number of the initial cluster (K). Moreover, we discuss whether the number of entries in log files affects the accuracy and detection rate of the system or not. Therefore, our implementation and experimental results describe the appropriate number of cluster and the proper amount of entries in log files. Finally, we show the result of our experiments with accuracy rate and number of initial cluster (K) graph, ROC curve and detection rate and false alarm rate table.**

Keywords*—Log analysis, K-means algorithm, Hadoop, Mahout, Distributed log analysis, KDD'99, Security, Intrusion Detection System, IDS

I. Introduction

In this paper, we determine the appropriate number of initial cluster and suitable number of entries for applying K-means algorithm to a TCPdump data set (obtained from KDD'99). With the appropriate number of initial cluster, K-means algorithm can efficiently cluster the data. In addition, the suitable number of entries means the system can efficiently analyze data in few iterations. Our analysis provides useful configurations for analysts that have applied Apache Mahout/Hadoop to TCPdump data. With data properly partitioned and appropriate number of initial cluster, our experiments show that large log data can be analyzed efficiently in limited amount of time.

Log Files are data containing list of events and activities that occur in the system. They are generated when events occur in the system. In large-scale systems, such as distributed systems, clusters and grid systems, huge amount of log data are generated in real time. Their humungous size is one of the vulnerabilities of the system because standalone log file analyzers, also known as intrusion detection systems, cannot analyze them all. Even though standalone log analyzers can analyze all log files, the results might not be accurate.

Although, standalone log analyzers cannot analyze huge amount of log files, we can still utilize these log files in a trace analysis. Trace back is an investigation not a prevention method. Hence, analysis of log files can be more beneficial than trace back.

In this paper, we apply Hadoop, which is a framework for the distributed system for large-scale log analysis. The log files we used are KDD'99 [1] data set. KDD'99 is a data set in the field of intrusion detection system. KDD'99 data set is a large-scale log files, it has approximately five million entries. Moreover, KDD'99 is commonly used in many intrusion detection research.

In [21], a group of researchers proposed mixed intrusion detection system which consists of misuse and anomaly detection. K-means algorithm is applied for the anomaly detection module. In their experiments, they generate four experiment data sets. Each experiment data set has two thousand and one hundred entries. Each data set contains two thousand entries of normal data. The rest is intrusion data. Their results show that K-means algorithm has high detection rate for a single intrusion and has low detection rate for multiple intrusions. In contrast, KD algorithm, which is their improved K-means algorithm has high detection rate for either single intrusion or multiple intrusions. However, their experiment data set is only two thousand and one hundred entries.

In [22], a group of researchers proposed distributed intrusion detection system (IDS) using KDD'99 data set as an example. With the distributed IDS, they applied K-means algorithm based on distributed model for clustering. In their experiments, they resampled data sets, which consisted of 1 to 1.5% of intrusion data and 98.5 to 99% of normal data. Finally, they evaluated their experiments by plotting Receiver Operating Characteristic (ROC) curve [7], [13].

In [17], they proposed intrusion detection model using Fuzzy C-means Clustering and used KDD'99 data set. The result of their experiments is effective for anomaly detection. Their results showed that if the number of clustering centers increased, the false alarm rate and detection rate will also increase.

Clustering algorithms are commonly applied to the field of

978-1-4799-0545-4/13/$31.00 ©2013 IEEE

intrusion detection system. However, Classification algorithms are also applied too. In [11], [15], a group of researchers proposed to use classification methods for intrusion detection. In [15], they studied and compared three classification techniques for intrusion detection. There are C5.0 Decision Tree, Ripper Rule and Support Vector Machine (SVM). They researched and compared which algorithm in these three algorithms is the most efficient technique. For the evaluation, KDD'99 data set is used. Their experiments indicated that C5.0 Decision Tree produced the accurate results which the detection rate higher than 96%. Nonetheless, clustering techniques are properly used for anomaly detection more than the classification techniques.

In [11], they proposed a new approach by using fuzzy neural network and Support Vector Machine (SVM) classification algorithms to improve the detection rate of the system. Moreover, K-Means clustering is applied at the first step of their approach. KDD'99 data set is used for their experiments. Their experiments show that their approach got the better results which the accuracy higher than 97% of all attack types (DoS, PROBE, U2R, R2L) in KDD'99 data set.

The rest of this paper is organized as follows: Section 2 is the related works. Section 3 describes our implementation. Section 4 discusses the results of our experiments and the last section is the conclusion of our works.

II. Related Works

This section is divided into 4 parts. They are K-means Clustering Algorithm, Confusion Matrix, Apache Hadoop and its projects, Receiver Operating Characteristic (ROC) Curve and Silhouette index.

A. K-means Clustering Algorithm

K-means algorithm [12], [14], [19], [20] is a well-known clustering algorithm. The main objective of clustering algorithm is to partition data with the similar characteristics into groups. K-means algorithm is the commonly used clustering algorithm. The main idea of K-means can be described by the following equation:

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} distance^2(x, m_i)$$

Assuming that x is an observation dataset $(x_1, x_2, x_3, ..., x_n)$ in cluster C_i and mi is the center point (centroid) of C_i cluster, the n observations will be partitioned into predefined k clusters $(k \leq n)$. This equation calculates distance (E) from each observation to the centroid of each cluster. Each observation will be assigned to the nearest cluster centroid measured by distance. The result after partitioning is the set of cluster $(C_1, C_2, C_3, ..., C_k)$.

After distances are calculated and each observation is assigned into cluster, the centroid of each cluster is recomputed. The process is repeated until the centroid of each cluster stop changing.

B. Confusion Matrix

Confusion matrix [3] is a specific table containing 4 values which are true positives, true negatives, false positives and false negatives. These values represent the performance of the prediction test of an algorithm. Each value can be described by comparing the predicted labels to the real labels as follows:

1) True Positives (TP): is the percentage of attack patterns which are correctly identified as the intrusion.

2) True Negatives (TN): is the percentage of normal patterns which are correctly identified as the normal activities.

3) False Positives (FP): is the percentage of normal patterns which are incorrectly identified as the intrusion.

4) False Negatives (FN): is the percentage of attack patterns which are incorrectly identified as the normal activities.

With 4 values from the confusion matrix, we can derive other statistical measure values, such as true positive rate (TPR) or sensitivity, false positive rate, true negative rate (TNR) or specificity, false negative rate and accuracy. These statistical measure values are represented as the following equation [8]:

1) True positive rate (TPR) or sensitivity: is the proportion of attack patterns which are correctly identified as the intrusion to the all exactly attack patterns.

$$TPR(Sensitivity) = \frac{TP}{(TP + FN)}$$

2) False positive rate (FPR): is the proportion of normal patterns which are incorrectly identified as the intrusion to the all exactly normal patterns.

$$FPR = \frac{FP}{(FP + TN)}$$

3) True negative rate (TNR) or specificity: is the proportion of normal patterns which are correctly identified as the normal activities to the all exactly normal patterns.

$$TNR(Specificity) = \frac{TN}{(TN + FP)}$$

4) False negative rate (FNR): is the proportion of attack patterns which are incorrectly identified as the normal activities to the all exactly attack patterns.

$$FNR = \frac{FN}{(FN + TP)}$$

5) Accuracy: is the proportion of all the correctly identified patterns to all the correctly and incorrectly identified patterns.

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

6) Positive predictive value (Detection rate): is the proportion of attack patterns which are correctly identified as the intrusion to all the predicted attack patterns.

$$PPV = \frac{TP}{(TP + FP)}$$

The performance of the system or the algorithm can be represented by the confusion matrix and other statistical measure values. Detection rate and false positive rate (false alarm rate) are commonly used to represent the performance of the system. True positive rate and false positive rate are also used to represent the performance of a binary classifier system by a Receiver Operating Characteristic (ROC) or ROC Curve [7], [13].

C. Apache Hadoop and its projects

Apache Hadoop [10] is a framework for the distributed system and cluster for processing large set of data using a MapReduce programming model. MapReduce [5] is a programming paradigm for distributed processing with large datasets proposed by Google. Its concept is to process large datasets with thousands of machines in clusters instead a single server. Hadoop provides high availability, fault tolerance and scalability with three subprojects: Hadoop Common, Hadoop Distributed File System (HDFS) and Hadoop MapReduce.

Apache Mahout [2], [6] is one of an Apache project that provides scalable machine learning algorithms such as clustering, classification, Association rule mining and batch based collaborative filtering etc. These algorithms are implemented on a top of Hadoop using MapReduce Programming model.

D. Receiver Operating Characteristic (ROC) Curve

A Receiver Operating Characteristics (ROC) curve [7], [13] is a graph which depicts the performance of classifiers system. The graph is plotted by the fraction of true positive rate (TPR) and false positive rate (FPR). With ROC graph, we can determine the appropriate threshold or model the classifier system. Even though ROC curve is commonly used in medicine, machine learning and other classifier systems, in this paper, we apply ROC curve to determine the performance model of K-means algorithm which is a clustering algorithm. By applying ROC for K-means, we use various number of K for clustering and plot the result of each K in a graph.

E. Silhouette Index

Silhouette index [9], [16], [18] is one of validate methods for clustering techniques. It represents the quality of the cluster algorithm. It indicates that the data points are properly clustered or not. Silhouette index (S_i) equation is defined in expression (1) [9], [16], [18]:

$$S_i^j = \frac{b_i^j - a_i^j}{max(a_i^j, b_i^j)} \tag{1}$$

From the expression (1), S_i^j value is between -1 and 1. For value 1 of S_i^j, It indicates that data points are properly grouped. The data points are closer to other points in their own cluster than other points in neighbor clusters. In contrast, if S_i^j value is equal to -1, it indicates that data points are closer to other neighbor clusters than their own clusters. So, Silhouette index is an evaluation index which used to compare the result of clustering methods that the data points are properly partitioned or not.

As shown in the expression (1), Silhouette index is calculated from a_i and b_i variables. a_i variable is the average distance from the i-th data point to other data points in the same cluster. b_i variable is the average distance from the i-th data point to all data point in the nearest cluster. a_i and b_i is given by the following expression [9], [16], [18]:

$$a_i^j = \left(\frac{1}{m_j - 1}\right) \sum_{\substack{k=1 \\ k\neq i}}^{m_j} d(x_j^i, x_j^k), \quad i = 1, \ldots, m_j \tag{2}$$

$$b_i^j = \min_{\substack{n=1,\ldots,K \\ n\neq j}} \left(\frac{1}{m_n} \sum_{k=1}^{m_n} d(x_j^i, x_k^n)\right), \quad i = 1, \ldots, m_j \tag{3}$$

Assuming that x is an observation data set $(x_1, x_2, x_3, ..., x_n)$ and C is a set of cluster $(C_1, C_2, C_3, ..., C_K)$. Let d$(x_i, x_j)$ is the distance between two data points x_i and x_j. Assuming that we have K cluster, for each cluster is represents by C_j where $j = (1, 2, 3, ..., K)$. The number of members in each cluster represents by m_j where m_j is a number of members in the j-th cluster. The members of the j-th cluster is represents by $(x_1^j, x_2^j, x_3^j, ..., x_{m_j}^j)$

From the expression (2), a_i^j is the average distance from the i-th data points in the j-th cluster to others data points in the j-th cluster. So, a_i^j values of the j-th cluster will be calculated equal to the number of members in the j-th cluster (m_j) where $i = (1, 2, 3, ..., m_j)$. From the expression (3), b_i^j is the average distance from the i-th data point in the j-th cluster to all data point in the nearest cluster.

After a_i^j and b_i^j are calculated, Silhouette value (S_i^j) for the i-th data point in the j-th cluster is calculated followed by the expression (1). Then, Silhouette value for each data point in the j-th cluster is calculated. Therefore, Silhouette index of the j-th cluster is calculated by the following expression:

$$S_j = \frac{1}{m_j} \sum_{i=1}^{m_j} S_i^j \tag{4}$$

From the expression (4), Silhouette index for each cluster will be calculated. Then, the global Silhouette index is presented. It indicates the quality and the properly of the clustering results.

Davies-Bouldin index [4], [16] is another popular evaluation index for clustering. In [16], their experiments show that Silhouette index produces more accurate results than that of the Davie-Bouldin index. However, Silhouette index uses much more computation time than that of Davies-Bouldin index. Moreover, Silhouette index computation method is more complex than the Davies-Bouldin method.

III. Implementation

For implementation and experiments we use KDD'99 [1] data set as input data. KDD'99 data set is a raw TCP dump data which generated by MIT Lincoln Labs for nine weeks. Lincoln Labs simulates a typical U.S. Air Force local-area-network (LAN) for a true Air Force environment. Therefore, KDD'99 has approximately five million entries for the full data set and has approximately five hundred thousand entries for the ten-percentage subset. Each entry of data set consists of 41 attributes.

There are four main attack types in KDD'99 data set. They are denial-of-service (DOS)(e.g. SYN flood), unauthorized access from a remote machine (R2L) (e.g. guessing password), unauthorized access to local super user (root) privileges (U2R) (e.g. buffer overflow attacks), and surveillance and other probing (Probing) (e.g. port scanning).

One of the prominent of KDD'99 data set is that for each connection is labeled as either normal, or as an attack, with exactly one specific attack type. Therefore, there is only one label for each entry and this label falls into only one main category of attack.

Our implementation contains 4 main steps as follows:

1) Reducing the number of entries.
2) Encoding attributes as numbers.
3) Clustering with K-means algorithm based on Map-Reduce paradigm.
4) Performance Analysis.

A. *Reducing the number of entry*

A full data set of KDD'99 has approximately five million entries, each with the unique label. Therefore, we reduce the number of entries by first removing the duplicate entries. After this method, the number of entry is reduced. After the reduction, the dataset has approximately one million and seventy five thousand entries. However, KDD also distribute a smaller data set, namely the ten-percentage subset. This ten-percentage subset dataset of KDD'99 has approximately one hundred and fifty thousand entries. We will use this smaller dataset for our experiments first.

B. *Encoding to numbers*

The proper data format for clustering is number. Therefore, each attribute and each entry have to be converted to number. For each entry, Euclidean distance to the centroid will be calculated. To make it easier for the calculation, each attribute is multiplied with the constant number to increase the distance of each entry to the centroid. This encoding and multiplying process makes the clustering easier because of the increasing in the distance from each entry to each centroid.

C. *Clustering with K-means algorithm based on Map-Reduce paradigm*

After the preprocessing, we will get the proper format of data set and unique entries of data set with their labels. With these data in hand, we apply K-means algorithm based on Map-Reduce paradigm for clustering them. Then, the result of the clustering is analyzed using confusion matrix.

D. *Analysis clustering results*

After the clustering, we derive statistical measure values from the confusion matrix. Then, we analyze the relationship between the number of K for K-means clustering and accuracy of the result by that the number of K. Finally, the appropriate number of K and the proper amount of entry are presented.

IV. EXPERIMENTS

In this section, we discuss the result of K-mean algorithm and analysis of these results. We determine the appropriate number of K and the adequate iteration compare to the accuracy. Therefore, we show the results as follows:

1) Accuracy and the number of cluster
2) ROC graph
3) Detection rate and false alarm rate table.

After the discussion of these results, we will get the proper value of these values.

In this experiment, the tests are run on a 4-node hadoop cluster. Each node is running Ubuntu Server 11.10. powered by an Intel(R) Core(TM) i5-2500 CPU @ 3.30GHz with 4GB of RAM and Hadoop 0.20.2-cdh3u5 software distributed by Cloudera.

A. *Accuracy and the number of cluster*

The accuracy and the number of cluster is shown in Fig. 1. As shown in the picture, we generate five dataset by resampling from the ten-percentage subset of KDD'99. Each sampling data sets has twenty thousand entries. With these sampling data sets and ten-percentage subset of KDD'99, we cluster them by applying K-means algorithm based on Map-Reduced model on Apache Mahout/Hadoop framework. For each testing data set, we run K-means with number of K ranging from 2 to 50. For each number of K, the maximum number of iteration is 300 iterations. The results of Accuracy and the number of K for each data set are shown in Fig. 1.

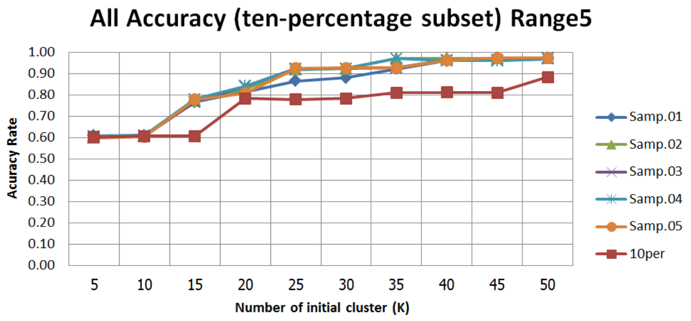

Fig. 1: All Accuracy of 10 percentage data set

According to the graph, each test data set has approximately the accuracy rate of 0.6 at the beginning. This value increases when increasing the number of K. We determine that the appropriate number of K is 25. At 25, there are four sampling data sets from five sampling data sets which has accuracy rate of 0.92 and another sampling data sets accuracy rate is approximately 0.86. In contrast, a ten-percentage subset of KDD'99 data sets has approximately the accuracy rate of 0.78.

According to the graph, the proper number of K is 25, but the accuracy rate decreases when the number of entries increases.

B. *ROC graph*

The ROC graph of the ten-percentage subset of KDD'99 is shown in Fig. 2. It shows the fraction of true positive rate (TPR) and false positive rate (FPR) of various numbers of clusters (K). In this plotting, we select only 10 numbers of K. There are K5, K10, K15, K20, K25, K30, K35, K40, K45 and K50. The value pairs of TPR and FPR of each selected number of K are shown in Table 1 and the ROC graph of these selected numbers of K is shown in Fig. 2.

According to the ROC graph, K50 is a proper threshold, which have high rate of TPR and low rate of FPR. K35, K40 and K45 are also proper groups. However, these numbers of K are too excessive. Therefore, K20, K25 and K30 are chosen to be a proper groups with the adequate number of K. As shown in the graph, these groups have similar values of TPR and FPR. Therefore, Table 1 shows the details information.

According to Table 1, K15 to K30 from all of sampling data sets have high rate of TPR and low rate of FPR except for the ten-percentage subset of KDD'99 data sets. If we consider these numbers of K and the value pairs of their results, K25

TABLE I: True Positive Rate (TPR) and False Positive Rate (FPR) of 10 percentage data set and its sampling data set

K	10percentage		10per (Sampling01)		10per (Sampling02)		10per (Sampling03)		10per (Sampling04)		10per (Sampling05)	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
5	0.00132	0.00034	0.01623	0.00405	0.00138	0.00008	0.00126	0.00000	0.00127	0.00033	0.00137	0.00008
10	0.01783	0.00198	0.01648	0.00215	0.01897	0.00158	0.01662	0.00182	0.01526	0.00124	0.01633	0.00125
15	0.01847	0.00203	0.96425	0.35824	0.96408	0.34898	0.96424	0.36230	0.96364	0.33468	0.96447	0.34435
20	0.96480	0.33265	0.95892	0.27807	0.96107	0.25710	0.95983	0.23659	0.95805	0.23150	0.96098	0.29193
25	0.96470	0.34075	0.95892	0.19336	0.95001	0.09993	0.94862	0.09014	0.94699	0.09601	0.95375	0.09174
30	0.96587	0.33353	0.94942	0.16356	0.94838	0.08938	0.94988	0.09031	0.94559	0.08777	0.95350	0.08824
35	0.96061	0.28578	0.95005	0.09404	0.93055	0.00183	0.94963	0.08616	0.93148	0.00115	0.95350	0.08982
40	0.96059	0.28424	0.94929	0.02956	0.93984	0.00440	0.94875	0.02629	0.94444	0.02794	0.95250	0.03039
45	0.96078	0.28605	0.93902	0.00471	0.93921	0.00440	0.94862	0.02795	0.94495	0.02901	0.94327	0.00509
50	0.94845	0.15598	0.93813	0.00421	0.93670	0.00357	0.94560	0.00862	0.92601	0.00148	0.94278	0.00467

gives the best results regardless of the data sets are being sampled or not. For other data set, Silhouette index or other evaluation index can be applied to indicate the appropriate number of K instead of ROC Curve.

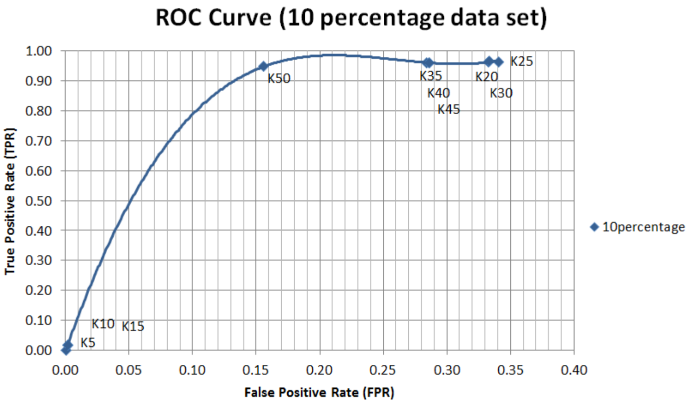

Fig. 2: ROC Curve of 10 percentage data set

C. Detection rate and false alarm rate table

The detection rate (PPV) and false alarm rate (FPR) table is shown in Table 2. As shown in the table, we have two main data sets and we generate five random sampling sets from each main data set. Thus, we have totally twelve test data sets. For each test data set, which is generated from the ten-percentage subset of KDD'99 data sets, it has twenty thousand entries per set. For each test data set generated from the full data set of KDD'99, it has two hundred thousand entries per set.

TABLE II: Detection rate and False alarm rate at K25 of all data set

Data set (K25)	Detection Rate (PPV)	False Alarm Rate (FPR)
10 percentage	0.65055	0.34075
10per.Samp.01	0.76358	0.19336
10per.Samp.02	0.86278	0.09993
10per.Samp.03	0.87390	0.09014
10per.Samp.04	0.86476	0.09601
10per.Samp.05	0.87439	0.09174
Full Data Set	0.73099	0.00023
Full.Samp.01	0.68902	0.00034
Full.Samp.02	0.52715	0.28298
Full.Samp.03	0.52602	0.28021
Full.Samp.04	0.52602	0.28021
Full.Samp.05	0.51017	0.29592

According to the Table 2, detection rate and false alarm rate with K25 vary depending on the number of entries of the test data sets. For the resampling data set of the ten-percentage subset (twenty thousand entries), its detection rate is approximately 0.86 and false alarm rate is approximately 0.09. For the ten-percentage subset of KDD'99 data sets (approximately one hundred fifty thousand entries), its detection rate drops to approximately 0.65 and false alarm rate increases to 0.34

For the sampling data sets from full data set of KDD'99 (two hundred thousand entries), its detection rate is approximately 0.52 and false alarm rate is approximately 0.28. For the last test data set, full data set of KDD'99 has approximately five million entries. Its detection rate is approximately 0.73 with very low false alarm rate.

In our experiment, we learned that accuracy varies depending on the number of K for K-means clustering. For the same amount of data set, a large number of K will result the higher accuracy than that of the small number of K. Also the true positive rate increases with the large number of K. Moreover, the execution time increases if we use the larger number of K. However, the appropriate number of K for our experimental data set is 25.

From our experiments, they show that the amount of entry has an effect on the detection rate and false alarm rate. The appropriate amount of entries which yield high detection rate and low false alarm rate are twenty thousand entries. Approximately, one hundred fifty thousand entries also yield the acceptable detection rate and false alarm rate.

V. Conclusion and Future Work

In this paper, we apply Hadoop for large-scale log analysis and we use KDD'1999 data set as our test data. Our main objective is to efficiently cluster these data by proposing the appropriate number of cluster and proper amount of entries for K-means clustering based on the Apache Mahout/Hadoop framework. With Hadoop and our preprocessing, our system can support large log files. Our experiments use the full data set of KDD'99.

Our experiments show that the appropriate number of initial cluster (K) for our experimental data set is 25 and the proper amount of entries in log file is 20,000 entries. However, approximately one hundred fifty thousand entries also produce the acceptable detection rate and false alarm rate.

In the future, the adequate iteration and the accuracy of that iteration will be presented. Moreover, we will determine the appropriate preprocess method to detect specific intrusion data. Finally, we aim to extract a new knowledge valuable for generating firewall rules.

References

[1] (2013) ACM KDD CUP [Online]. http://www.sigkdd.org/kdd-cup-1999-computer-network-intrusion-detection.

[2] (2013) apache mahout [Online]. http://en.wikipedia.org/wiki/Apache_Mahout.

[3] (2013) confusion matrix [Online]. http://en.wikipedia.org/wiki/Confusion_matrix.

[4] (2013) DaviesBouldin index [Online]. http://en.wikipedia.org/wiki/DaviesBouldin_index,.

[5] (2013) MapReduce [Online]. http://en.wikipedia.org/wiki/Mapreduce.

[6] (2013) overview - apache mahout - apache software foundation [Online]. https://cwiki.apache.org/confluence/display/MAHOUT/Overview.

[7] (2013) receiver operating characteristic [Online]. http://en.wikipedia.org/wiki/Receiver_operating_characteristic.

[8] (2013) sensitivity and specificity [Online]. http://en.wikipedia.org/wiki/Sensitivity_and_specificity.

[9] (2013) silhouette (clustering) - wikipedia, the free encyclopedia [Online]. http://en.wikipedia.org/wiki/Silhouette_(clustering).

[10] (2013) welcome to apache hadoop! [Online]. http://hadoop.apache.org/.

[11] A. Chandrasekhar and K. Raghuveer. Intrusion detection technique by using k-means, fuzzy neural network and svm classifiers. In *Computer Communication and Informatics (ICCCI), 2013 International Conference on*, pages 1–7, 2013.

[12] D. Denatious and A. John. Survey on data mining techniques to enhance intrusion detection. In *Computer Communication and Informatics (ICCCI), 2012 International Conference on*, pages 1 –5, jan. 2012.

[13] T. Fawcett. An introduction to roc analysis. *Pattern Recogn. Lett.*, 27(8):861–874, June 2006.

[14] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Clustering algorithms and validity measures. In *Scientific and Statistical Database Management, 2001. SSDBM 2001. Proceedings. Thirteenth International Conference on*, pages 3 –22, 2001.

[15] R. Naidu and P. Avadhani. A comparison of data mining techniques for intrusion detection. In *Advanced Communication Control and Computing Technologies (ICACCCT), 2012 IEEE International Conference on*, pages 41–44, 2012.

[16] S. Petrovi. A comparison between the silhouette index and the davies-bouldin index in labelling ids clusters, 2006.

[17] W. Ren, J. Cao, and X. Wu. Application of network intrusion detection based on fuzzy c-means clustering algorithm. In *Intelligent Information Technology Application, 2009. IITA 2009. Third International Symposium on*, volume 3, pages 19 –22, nov. 2009.

[18] P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(0):53 – 65, 1987.

[19] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[20] Wikipedia. (2013) *k*-means clustering — Wikipedia, the free encyclopedia [Online]. http://en.wikipedia.org/wiki/K-means_clustering.

[21] C. Zhang, G. Zhang, and S. Sun. A mixed unsupervised clustering-based intrusion detection model. In *Genetic and Evolutionary Computing, 2009. WGEC '09. 3rd International Conference on*, pages 426 –428, oct. 2009.

[22] Y.-F. Zhang, Z.-Y. Xiong, and X.-Q. Wang. Distributed intrusion detection based on clustering. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, volume 4, pages 2379 –2383 Vol. 4, aug. 2005.