Multi-density Clustering Algorithm for Anomaly Detection Using KDD'99 Dataset

Conference Paper *in* Communications in Computer and Information Science · July 2011

DOI: 10.1007/978-3-642-22709-7_60 · Source: DBLP

3 authors:

Kumar Santosh
Indian Institute of Technology Guwahati
13 PUBLICATIONS **38** CITATIONS
SEE PROFILE

Sumit Kumar
Samsung R&D Institute, INDIA
12 PUBLICATIONS **10** CITATIONS
SEE PROFILE

Sukumar Nandi
Indian Institute of Technology Guwahati
342 PUBLICATIONS **2,709** CITATIONS
SEE PROFILE

Multi-density Clustering Algorithm for Anomaly Detection Using KDD’99 Dataset

Santosh Kumar, Sumit Kumar, and Sukumar Nandi

Department of Computer Science and Engineering
Indian Institute of Technology Guwahati
Guwahati, India
{santosh.kr,sumit.kr,sukumar}@iitg.ernet.in

Abstract. Anomaly detection is currently an important and active research problem in many fields and involved in numerous application. Handle huge amount of data or traffic over the network is most challenge full task in area of Intrusion Detection System to identify the intrusion by analyzing network traffic. So we have required the some efficient technique for analyze the anomaly from network traffic which have good detection rate with less false alarm and it should be also time efficient. Motivation by above, in this paper we present a Multi-density Clustering Algorithm for anomaly detection (MCAD) over huge network traffic (Offline statistical traffic). In this approach we have improved the Birch Clustering [1] index problem with ADWICE (Anomaly detection with fast Incremental Clustering) [2] model using grid index. We have used the Intra cluster distance parameter property which can improve the quality of cluster in respect of outliers by the average intra cluster distance reduction. So in this approach rather than threshold concept at insertion of data point in the cluster we have used the cluster quality indices for insert a data point in the cluster and checked it is being optimized or not. The method is verified by experimental of proposed approach on KDD’99 [3] data set which is standard off line data set. Experimental results illustrate better false alarm detection rate and time efficiency by using proposed MCAD approach.

Keywords: Anomaly Detection, k-mean clustering, ADWICE model of clustering, BIRCH model of clustering, MCAD clustering.

1 Introduction

A network intrusion attack can be any use of network that compromises its stability or the security information that is stored on computers connected to it. A very wide range of activity falls under this definition, including attempts to destabilize the network as a whole, gain unauthorized access to file or privilege, or simply mishandling and misuse of software. Intrusion detection is the process of identifying and responding the malicious activity targeted at computing and networking resources. Intrusion detection systems are software or hardware product that monitor and analyze network. In particular Network based intrusion detection system called row data packets from the network and carefully analyze for abnormal or anomaly packets thereby detecting

A. Abraham et al. (Eds.): ACC 2011, Part I, CCIS 190, pp. 619–630, 2011.

security violations. Unlike host based IDS [4], network based IDS [5] protects a group of system by generalizing the security concept to a network.

In anomaly detection models the behavior of the system with a profile and any deviation from the known pattern is considered as intrusion. There are mainly three types of Anomaly detection techniques according the data labels, namely as Supervised anomaly detection [6], Unsupervised anomaly detection [6] and Semi-supervised anomaly detection [6]. In supervised anomaly detection training data set are labeled as normal and abnormal or we can build a model with both type of data set. A classifier model in which only normal data set used for the training is called Semi-supervised anomaly detection. While in Unsupervised anomaly detection the training data instances are not labeled so it is less complex. In unsupervised assumption is made that normal data are larger in comparison of abnormal or anomaly data.

Anomaly detection still faces many challenges, where one of the most important is the relatively high false alarm. Recently many data mining techniques used for the anomaly detection, some of them are: Machine learning based [7], decision tree based [8], self-organizing map based, K-mean clustering based [9], Birch clustering based, fuzzy c-Mean clustering [10] and finite automata based etc. We have proposed a Multi-density clustering based approach for anomaly detection, which is an improvement of ADWICE model of Birch clustering. In this approach we have used an average intra cluster distance in which rather than threshold concept at insertion of data point cluster we have used the cluster quality indices for insert the data point into the cluster and check the optimality for same. For experiment of proposed model we have used the KDD'99 standard data set for training and testing data.

2 Literature Survey

In this part we have to explain some basic knowledge of clustering to make obvious sense of problem statement and description of aim of the paper. Literature survey follows as:

2.1 Clustering

A process of grouping a set of physical or abstract objects into classes of similar objects is called clustering and a cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. Given two objects, represented as normalized feature vectors with continuous roughly linear variables, the similarities can be computed by considering the geometrical distance between the two vectors. A common distance measure is the well known Euclidian distance [1]. There are various types of clustering as:

2.1.1 k-Mean Clustering [9]

K-mean is partitioning clustering. It divides the data points into k clusters. In this clustering randomly choose k data instance from data points and make them initial cluster center after that assign the points nearest of the cluster center the replace each center with mean f the points around the cluster center. Repeat above process until there is no further updating of cluster center. The advantages of K-mean clustering are its scalability and its time complexity $O(nkt)$. Where n denotes the number of points, While k is number of partitions and t is a number of iterations. The disadvantages of K-mean clustering are, it is not able to find non convex cluster and defining number k cluster before clustering and obtaining k is NP hard.

2.1.2 BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies) Cluster [1]

It is designed for large amount of numerical data by integration of hierarchical clustering and other iterative clustering. It overcomes the two problem of hierarchical clustering by making it scalable and making it able to undo what was previously done. BIRCH store a compact summarization in from of clustering feature and thus reduce the problem of clustering the original data points into one of clustering the set of summaries, which is much smaller than the original dataset. Clustering decisions in BIRCH are made without scanning all data points or all currently existing clusters and thus it is said to be incremental. There is a Database oriented constraint in BIRCH that the amount of memory available is limited where as dataset can be arbitrary large mean that memory available can be 20% of the database. The advantages of BIRCH clustering's fast enough due to no I/O operations are needed. In this clustering we don't have to work on entire data points rather than we have to work on sub clusters and more accurate because more outlier can be eliminated. The time complexity of BIRCH clustering is $O(n)$. The disadvantages of this clustering's, this clustering is not suitable for multi-density cluster. It keeps same threshold for the entire sub cluster for insertion of points whether cluster are small and dense or sparse and big.

3 Problem Definition and Proposed Algorithm

Kalle Burbeck and Simin Nadjm-Tehrani presents an ADWICE model [2] which used the first phase of the existing BIRCH clustering framework to implement fast, scalable and adaptive pure anomaly detection. In this model ADWICE they used BIRCH clustering in which only cluster feature of the data points of clusters are stored and it used grid index to detect the anomaly. It works on the concept of pure anomaly detection based system in which it form cluster of normal packets while training a model. According to this model we had to work on cluster features rather than data points. The distance between data point and a cluster is calculated from Euclidean distance between data point and the centroid of the cluster and the distance between two clusters can be calculated from the Euclidean distance between the centroids. Each cluster of the leaf node can absorb new data point if Euclidean distance between data point and centroid is less than threshold requirement.

There are three basic principle of ADWICE model for learning or adapting.

a) If no cluster is close enough to absorb the data point then data point vector v_i is inserted into the model as a new cluster. If there does not exist a leaf subspace in which the new cluster fits, the new leaf is created. However, there is no need of any additional update of the tree, since higher up nodes do not contain any summary of data below.

b) When the closet cluster absorbs to v_i , its centroid is updated accordingly. This may cause the cluster to move in space. A cluster may potentially move outside its current subspace. In this case, the cluster is removed from its current leaf and inserted into a new tree from the root, since the path all the way up to the root may have changed. It the cluster was the only one in the original leaf, the leaf itself is removed to keep unused subspace without any leaf representations.

c) If cluster is removed or forgotten the index is only changed if the leaf is now empty in which case the leaf of the removed cluster will also removed.

As ADWICE model uses BIRCH clustering for cluster the data and BIRCH cluster itself unable to hold multi-density cluster as it use distance based measures to determine that whether to include data point in the cluster or not. At the same time it use same threshold for forming all cluster whether the cluster is sparse and big cluster or the cluster is small and dense. The BIRCH cluster uses same threshold while insertion of a point and then during merging of cluster it increment same threshold so lots of points which should not be included in the cluster are being included.

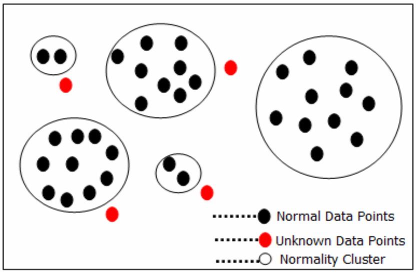


Fig. 1. ADWICE model for anomaly

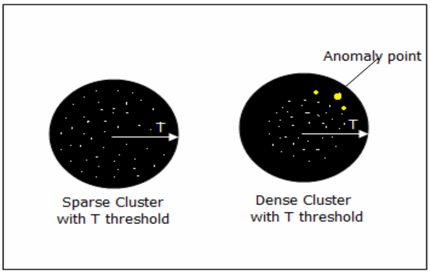

Anomaly point
T
T
Sparse Cluster with T threshold
Dense Cluster with T threshold

Fig. 2. ADWICE model thresholds

We got motivation form above disadvantages of ADWICE model consequently we are proposing a Multi-density Clustering Anomaly Detection (MCAD) algorithm for pure anomaly detection to reduced the diameter of dense and small cluster and keeps the advantages of BIRCH cluster. Our algorithm remains linear using summarization technique of cluster. According to Ying zhao and George Karypsis in Hierarchical Clustering Algorithms for Document Datasets [12], the average intra-cluster distance is the parameter which can be used to make the quality cluster. We have used the property of cluster quality improvement in which the cluster quality will improve however the average intra cluster distance reduces. So we have used cluster quality indices to insert a point in the cluster rather than threshold concept of insertion point in the cluster and checked whether it is being optimized or not. The proposed training and testing as followed.

3.1 Training Steps for Proposed Anomaly Detection Algorithm

INPUT: NC Number of clusters, Training data packets, $\alpha - cons\tan t$ which is a multiple of cluster quality indices ratio of intra-cluster distance and inter-cluster distance used while merging step.

OUTPUT: Trained model with NC number of clusters.

- **Step1:** insert (data packets m)
- **Step2:** Descend cluster feature (CF) tree if average intra-cluster distance reduces or have minimum change
- **Step3:** if m optimizes leaf node average intra-cluster distance
- **Step4:** then add cluster feature (CF) packet to the leaf node
- **Step5:** else
- **Step6:** if $(leafnode \prec BranchingFactor)$
- **Step7:** add it next to leaf where it reduces AID minimum.
- **Step8:** update the CF tree up to parent node
- **Step9:** else
- **Step10:** split (leaf node, m)
- **Step11:** repeat step 1-10 till number of node equals to N
- **Step12:** rebuild tree
- **Step13:** traverse from left to right
- **Step14:** merge the cluster represented by node if average intra-cluster distance doesn't increases by $\alpha - cons\tan t$
- **Step15:**if $\left(size_{initial} \pmod{el} = size_{final} \pmod{el}\right)$
- **Step16:** increase $\alpha - cons\tan t$ repeat step 12-15.
- **Step17:** else repeat step 1-10.

Details of some above steps are given here as follows:

Step1: Insertion: For inserting m points into cluster feature tree we are looking for cluster indices quality which means that cluster becomes better quality if average intra-cluster distance decreases. So if point's m optimizes the average intra-cluster distance then it should be included in the cluster.

Step2: Identify the appropriate leaf: Starting from the root node we recursively descend the CF tree where it optimizes the average intra-cluster distance or if it does not reduces the average intra-cluster distance the we will looked for child node to which it have done minimum changes in average intra-cluster distance after merging so it will avoid the condition that if points are equal distance to the two cluster where it to be proceed.

Step3: Modifying the leaf and path: Reaching at the leaf node, find out the leaf entry which is being optimized and update the CF tree. If none of the leaf node is being optimized then we will add it besides the leaf entry to which it is nearest and if the number of leaf entry are lesser than branching factor of tree then it is nearest otherwise we had to split the nodes and modify up to the parent node and check out for

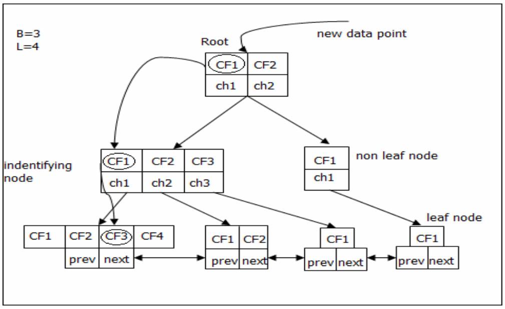


Fig. 3. Optimization of node in case of threshold and intra clustering distance

them also if the number of entry exceed branching factor than splitting occur at parent node also. Splitting occurs by taking the two farthest node of cluster and merging other node according to their closeness so after insertion of point we had to modify up to parent node of cluster summary, fig.4 shows the splitting of leaf node.

Step4: Rebuilding the tree: When the number of nodes representing each cluster reaches at maximum number of cluster then we had to rebuild the tree. For rebuilding the tree when we have single point as a cluster representative, then we cannot directly merge the closest point as though they are inter-relatively closer but rather than that point can be far apart into the overall scenario. So that for merge firstly we have required threshold parameter when we have only a single point for merging. This threshold is calculated by the calculating of average distance of the closest node center and to search closed node we had to check next node and previous node. After putting this threshold we assured that at least few points merged in to the cluster. Fig.5 shows the splitting of parent node and modified up to parent node.

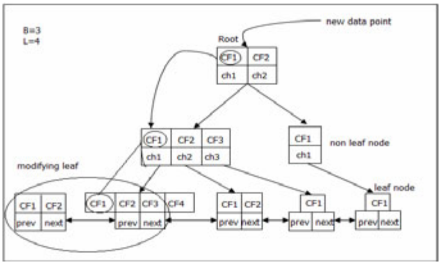


Fig. 4. Splitting of leaf node

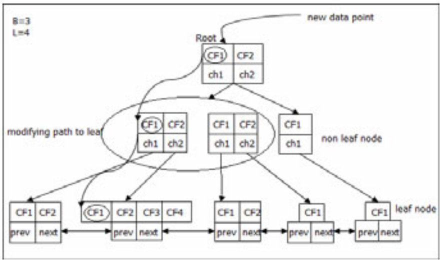
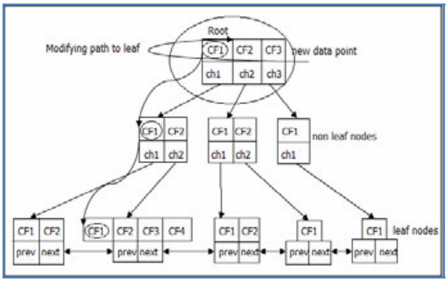


Fig. 5. Splitting of parent node

Step5: Identifying the proper child while descending from root node: In ADWICE model while descending from root node we have come across the problem where, the root node has equal distance from two child cluster. This problem we have solved in our proposed algorithm by using a condition that the point will goes to that cluster where it will increase lesser intra cluster distance, fig.7 shows the same condition where the point “m” has equal distance from cluster1 (left cluster) and cluster2.

According to ADWICE model the points goes to cluster2 (which is a error in ADWICE model) and according to our proposed MACD algorithm it will goes to cluster1.



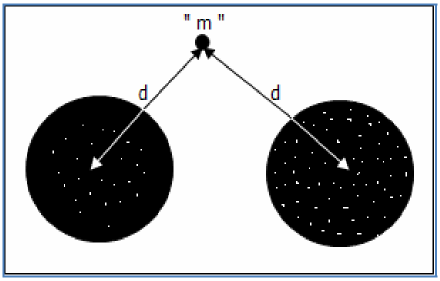
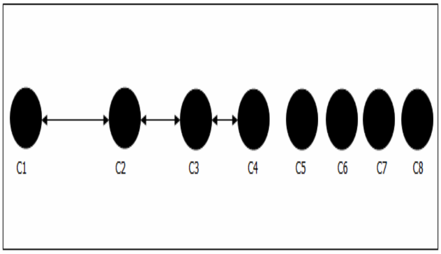


Fig. 6. Modification up to parent node

Fig. 7. Point m same distance from cluster 1 and 2

Step6: α**(Avg. intra cluster distance)**$_{\textbf{initial}}$ **>(Avg. intra cluster distance)**$_{\textbf{final}}$ **:** In ADWICE model if the cluster is far from denser and smaller cluster although it is within the range of threshold then it will not be merged when it has same threshold for all merging cluster whether it is bigger cluster or it is smaller cluster. For solving the problem we have used a condition where α(Avg. intra cluster distance) $_{\text{initial}}$ > (Avg. intra cluster distance) $_{\text{final}}$ used for cluster merging (not for direct merging). In this condition closest cluster merging if clusters are inter-relatively closer but and in overall scenario if they are far apart of it then they will not be merged. Figure8 shows C2 cluster is closer to C3 cluster but it will not merged to any cluster and if we increases the intra-cluster distance between cluster C2 and cluster C3 to a large ratio then it resulting as decreasing the cluster quality.



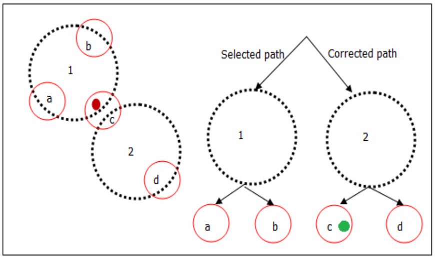


Fig. 8. Intracluster distance comparison

Fig. 9. Indexing of testing point in MCAD

Step7: Solving the Grid indexing problem of BIRCH and ADWICE model: BIRCH and ADWICE both model of clustering have used grid indexes for indexing the testing data points which suffers from high computational complexity and time complexity. The proposed model is based on intra clustering and inters clustering distance of each data point. So the proposed model doesn't suffer from complexity problem. Figure 9 shows that the point should goes to node2 but it goes to node1 at left.

3.2 Proposed Testing Algorithm for Anomaly Detection

After build a training model for anomaly detection, we have to test this training algorithm by following testing algorithm for anomaly detection algorithm. In the testing algorithm we have started the testing a point from the root node, if the testing point reduced the average intra cluster distance of parent node then we go ahead further for testing. If testing point does not optimizes any of the descendent node then it declared as anomaly point and if optimizes the leaf node then it declared as normal point. The algorithm shows the important steps of the testing algorithm.

Input: Clusters construct by training model.
Output: Decision on testing data points as anomaly or normal.

- **Step1:** Insert testing data
- **Step2:** Descend CF tree if the average intra cluster distance reduces or have a minimum change
- **Step3:** if a testing data point m optimizes the leaf node by reducing the average intra cluster distance then testing data point is normal.
- **Step4:** else if distance from center radius of cluster is greater than the average intra cluster distance then the testing data point is attack.

4 Experiment and Results

In order to estimate the performance of MCAD algorithm in anomaly detection, the algorithm is tested based on the KDD'99 data set [3] and compared with the traditional ADWICE and BIRCH algorithms for intrusion detection.

4.1 Experimental Data

KDD'99 data set have 5,000,000 records altogether including mainly four intrusion sorts: Dos, R2L, U2R and Probe. Each intrusion sorts contains some different small sorts. This is to large number of initial data of training and testing set for processing into the proposed training and model. So we have chosen the 30% of total training and testing data set of KDD'99 data.

The training set consists of 97500 normal records and the testing set contains the 20500 records including 19447 normal records and 1025 abnormal records. The percentage of anomaly records in testing data is 5% which is far less than the normal data set. Table 1 shows the experimental abnormal data used for testing model.

Each record in the KDD'99 data set is a network linked record. Each link consists of 41 features containing 3 symbolic variables and the others which are numerical variables.

Table 1. Abnormal testing data set distribution

Dos	R2L	U2R	Probe
Neptune (220)	Phf (40)	Buffover (9)	Portsweep(95)
Smurf (146)	Multihop (70)		Ipsweep (40)
Teardrop (80)	Warezmaster (95)		Satan (200)
	Root-kit (30)		

There are different measurement standards for the different features. In order not to affect the clustering result, the attribute values of data need to be process. The processing includes two steps. Firstly, the method in accordance with the protocol layer division is adopted to realize transforming the symbolic variable to the numerical value. When the TCP, UDP and ICMP in the protocol attribute, they should be separately set as 1, 2 and 3. Then all numerical variables are standardized and normalized to the number of [0, 1]. The standard deviation transform is as follows

$$x'_{ik} = \frac{x_{ik} - x_k^{-}}{S_k} \tag{1}$$

Where x'_{ik} is the kth attribute value of the i^{th} record in the 30% data set of KDD'99. The sample typical value x_k^{-} and the standard deviation S_k are given as follows

$$x_k^{-} = \frac{1}{n}\sum_{i=1}^{n} x_{ik} \tag{2}$$

$$S_k = \left(\frac{1}{n-1} \sum_{i=1}^{n} \left(x'_{ik} - x_k^{-} \right)^2 \right)^{1/2} \tag{3}$$

4.2 Determination of Number of Cluster and Branching Factor

The number of clusters N is to be decided by the experiment. If we set N to be the number of training data presents, then it will be the case in which all cluster contains the unique data points and a model in which if testing normal data is different from the training normal data set, hence results as a large number of false positive. So the number of clusters N should be lesser than the number of training data points. If the number of clusters N set to be one then there will be only one cluster representing the training data set which results as low detection rate of anomaly. So we can conclude that the number of clusters depends on the distribution of data. We have experimented with N=9000 to 13000 clusters. At 12500 we got the better detection rate and comparably lesser false positive rate. Similarly the branching factor also increases the training and testing time. The parameter branching factor equal to the number of data points would make the tree flat completely and make the algorithm linear as opposed to algorithmic in time. We have chosen the branching factor as 18. Figure 10 shows the importance of number of clusters required for anomaly detection.

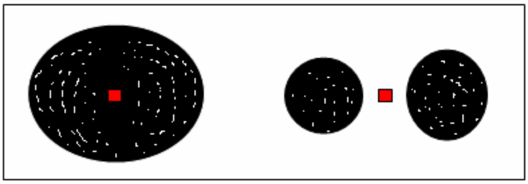


Fig. 10. Importance of numbers of clusters of Intrusion detection

4.3 Results

The proposed MCAD algorithm is realized by programming with java on PC (4 GB memory, Pentium 2.5 GHz CPU and Ubuntu10.4 operating system). The clustering MCAD algorithm is firstly trained with the training data set. Then the intrusion detection performance is evaluated in the testing data set. The detection rate and the false alarm rate adopted to interpret the performance of the algorithm. The detection rate denotes the percentage of the correctly detection intrusion number in all the recorded intrusion number in the testing data set. False alarm rate denotes the percentage of the number of normal data which is wrongly detected in all the normal number in a test set. Table 2 shows the detection rate of proposed algorithm with branching factor 18 along the number of clusters used in experiment.

Table 2. Attack and normal Detection rate of proposed algorithm using branching factor =18

Number of Clusters	Dos	Prob	U2R	R2L	Normal
9050	98.8	96.0	55.0	48.0	97.9
10000	98.9	96.2	61.2	52.1	97.4
11000	99.0	96.4	65.0	59.6	96.8
12000	99.2	97.0	72.8	69.2	95.2
13000	99.4	97.4	80.2	79.5	94.1

4.4 Comparison of Results with Other Clustering Algorithms

First we compare proposed MCAD algorithm space and time requirement with other clustering algorithms such as BIRCH, ADWICE and DBSCAN. Our algorithm gets less training space and training time among all the algorithms. Table 3 shows the results.

Table 3. Results of various cluster algorithms

	Various Clustering			
	MCAD	BRICH	ADWICE	DBSCAN
Training Space (k)	3298	5124	4425	13312
Training Time (ms)	4124	12923	5546	21478
Detection Time(ms)	264	947	341	1392

After comparison of time space and training time of our algorithm with other clustering algorithms we have compared the performance of our MCAD algorithm. Table 4 has shown the results of performance comparisons of various clustering algorithm.

Table 4. Performance comparison of proposed Clustering Algorithm

Attack	MCAD	BRICH	ADWICE	DBSCAN
Dos	99.2	97.8	98.3	96.3
Probe	97.0	95.5	96.0	93.8
U2R	72.8	81.2	81.1	56.2
R2L	69.2	70.1	70.8	46.2

5 Conclusion

In this paper we have proposed a novel clustering algorithm for anomaly detection. The algorithm achieved improved detection rate over some important clustering algorithm for anomaly detection. Results and Experimental part validate the proposed algorithm on KDD'99 Intrusion detection data set. We have also solved the BIRCH model indexing problem by including the cluster quality average intra cluster distance in our proposed algorithm which results as a conclusion that multi density clustering algorithm provide the better cluster as it make compact and small clusters.

References

1. Zhang, T., Ramakrishnan, R., Livny, M.: Birch: an efficient data clustering method for very large databases. In: SIGMOD Record 1996 ACM SIGMOD International Conference on Management of Data, pp. 103–114 (1996)
2. Burbeck, K., Nadjm-Tehrani, S.: ADWICE – anomaly detection with real-time incremental clustering. In: Park, C.-s., Chee, S. (eds.) ICISC 2004. LNCS, vol. 3506, pp. 407–424. Springer, Heidelberg (2005)
3. Mahoney, M.V., Chan, P.K.: An analysis of the KDD,99 darpa/lincoln laboratory evaluation data for network anomaly detection. In: Proceedings of 6th International Symposium on Recent Advances in Intrusion Detection, pp. 220–237 (2003)
4. Mukhrjee, B., Levitt, N.: Network Intrusion Detection. IEEE Networks 24, 26–29 (2005)
5. Han, H., Lu, X.L., Lu, J., Bo, C.: Data mining aided signature discovery in network-based intrusion detection system. ACM SIGOPS Operating System Review 36, 7–13 (2002)
6. Hilas, C.S., Mastorocostas, P.A.: An application of supervised and Unsupervised learning approaches to telecommunications fraud detection. ACM Journal of Knowledge-Based systems 21, 721–726 (2008)
7. Kumar, S., Nandi, S., Biswas, S.: Research and application of one-class small hypersphere Support Vector Machine for Network anomaly detection. In: The Third International Conference on Communication System and Networks (COMSNETS), pp. 1–4 (2011)
8. Yasami, Y., Mozaffari, S.P.: A novel unsupervised classification approach for network anomaly detection by k-means clustering and ID3 decision tree learning method. ACM Jounal of Supercomputing 53, 231–245 (2010)
9. Kanungo, T., Mount, D.M., Netanyahu, N.S.: An efficient k-Mean clustering Algorithm: Analysis and Implement. ACM/IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 881–892 (2002)
10. Hilal Inan, Z., Kuntalp, M.: A study on fuzzy C-mean clustering-based systems in automatic spike detection. ACM Journal of Computers in Biology and Medicine 37, 1160–1166 (2007)
11. Ester, M., Kriegel, H.-P., Sander, J.: A Desnsity-Based Algorithm for Discovering Clusters in Large Spatial Databased with Noise. In: Proceeding on 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226–231 (1996)
12. Zhao, Y., Karypis, G.: Criterion functions for document clustering, Experiments and Analysis. Technical report, 1–130 (2002)

Appendix

Intra and Inter-cluster distance: There is large difference between Intra-cluster and Inter-cluster distance. Inter-cluster distance measured by within-cluster sum of squares. Its measures cluster "compactness".
For one cluster r:

$$D_r = \sum_i \sum_j \left\| x_i - x_j \right\|^2$$

$$= 2n_r \sum_i \left\| x_i - x^- \right\|^2$$

For all k clusters:

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r$$

Clustering Features, Radius and Centroid of clusters: Clustering features (CF) includes the number of data points in a cluster (N), linear sum of data points (LS) and square sum of data points in a cluster (SS).

$$CF = \langle N, LS, SS \rangle$$

The centroid of a cluster given by:

$$X_0 = \int_{i=1}^{n} X_i$$

The radius of cluster given by:

$$R = \sum_{i=0}^{n} \left[\left(X_i - X_0 \right)^2 / n \right]$$

where i=1 to n

α **Value:** α represents the central value of cluster, can be calculated as

$$\alpha = \frac{1}{N} \sum_{i=1}^{r} X_i$$