ICIC Express Letters
Volume **14**, Number **10**, October **2020**

 ISSN 1881-803X
pp. **951–960**

DETECTING ANOMALY USING PARTITIONING CLUSTERING WITH MERGING

Fokrul Alom Mazarbhuiya[1], Mohammed Yahya Alzahrani[2] and Anjana Kakoti Mahanta[3]

[1]Department of Mathematics
School of Fundamental and Applied Sciences
Assam Don Bosco University
Tepesia, Assam 782402, India
fokrul_2005@yahoo.com

[2]Department of Information Technology
College of Computer Science and IT
Albaha University
Albaha 65525, Saudi Arabia
imohduni@gmail.com

[3]Department of Computer Sciences
Gauhati University
Jalukbari, Assam 781014, India
anjanagu@yahoo.co.in

Received December 2019; accepted March 2020

Abstract. *Nowadays, anomaly detection in data is an important field of research. A couple of methods have already been developed for this purpose, among which clustering-based anomaly detection remains an important one. Most methods using clustering approaches consider either numeric or categorical attributes of the data instance, but not both. However, in real life, most datasets are hybrids containing numeric, categorical, or mixed data. For such hybrid datasets, the typical distance formula does not work. This paper presents a unified distance formula for the clustering approach, in which the attributes may be numeric, categorical, or a mixture of both. Here, we propose an anomaly detection method based on both partitioning and hierarchical approaches. The algorithm proposed here is a modified version of k-means clustering algorithm in the sense that it also uses a suitably defined merge function. After every iteration the merge function is used to merge a pair of highly similar clusters to produce a larger cluster. The efficacy of our method is established using the complexity analysis and experimental results.*
Keywords: Intrusion detection, Information security, Outlier analysis, k-means algorithm, Object-cluster distance, Similarity measure, Merge function

1. **Introduction.** Due to the widespread use of computers and associated networks, it has become cheaper to store, transmit and process data. A huge amount of data that may contain valuable information is piling up on a daily basis. The problem of extracting such valuable knowledge is termed as *data mining*. So data mining is a method of discovering non-trivial and previously unknown information or patterns from huge datasets.

There are several approaches for this purpose, and clustering is one of them. Clustering is an unsupervised learning technique used to find patterns and data distributions in datasets. While it has been extensively studied in social science and psychology [1], it was made popular by the database community. There are two primary classes of clustering methods: partitioning and hierarchical.

DOI: 10.24507/icicel.14.10.951

The partitioning method divides the objects into predefined number clusters based on some criteria. Several algorithms have been developed for this task. The k-means algorithm is one of the most frequently used one. In [2], the author explained the k-means algorithm for clustering numerical data using the distance function. Meanwhile, in [3], the authors proposed a dynamical system-based method for clustering categorical data. In [4], the authors presented a partitioning method for clustering mixed data with numeric and categorical attributes. The merge function is an essential part of any agglomerative hierarchical clustering algorithm [5,6]. It is used to merge any pair of clusters with a high degree of similarity. Obviously, every merger produces a larger cluster.

An outlier is a data point that does not belong to any cluster. Extraction of outliers from large datasets is an emerging area of research in information security, and it has a number of applications, such as fraud, anomaly, and intrusion detection. In [7,8], the clustering-based outlier detection methods are discussed. Intrusion is an attempt to compromise the integrity, confidentiality, or availability of resources by accessing them in an illegitimate way. In [9,10], the authors used the fuzzy c-means clustering algorithm in intrusion detection. Intrusion detection techniques are broadly categorized into two types: (i) *anomaly detection techniques* and (ii) *signature recognition techniques* [11,12]. An *anomaly detection technique* is used to discover intrusions or misuses of networks and computers by monitoring and grouping system activities into normal or anomalous. The intrusion detection system (IDS) based on anomaly detection is termed the anomaly-based intrusion detection system. In [13], the authors proposed a k-means algorithm-based technique for traffic anomaly detection that uses the weighted Euclidean distance. Several works have been conducted in this direction; however, most of the methods used, consider only numeric attributes. Considering the numeric as well as the categorical attributes of the network data, a fuzzy c-means-based method for anomaly detection is discussed in [14] which have used distance on numeric attributes and dissimilarity on categorical attributes.

In this paper, we use both partitioning and hierarchical approaches for detecting anomaly in the network data. First, we define the data instance-cluster distance measure [4] in terms of both numeric and categorical attribute-cluster distance. Then, we define similarity between a pair of clusters in a similar fashion, and a merge function is defined in terms of the above similarity measure. Finally, a new algorithm for anomaly detection is proposed in this paper. The algorithm makes use of the approaches of both partitioning and agglomerative hierarchical clustering algorithms, and it is a modified version of k-means clustering algorithm, which also uses a suitably defined merge functions. The algorithm supplies a set of clusters, and the number of output clusters would be less than or equal to the number of randomly selected input clusters. The smaller cluster contains the outliers, and the extracted outliers are considered anomalies.

The paper is organized as follows. In Section 2, we briefly discuss the related works. In Section 3, we discuss the problem statement. The proposed algorithm for anomaly detection is discussed in Section 4. In Section 5, we discuss the time complexity of the algorithm. In Section 6, we discuss the experimental results. Finally, we conclude our paper with a brief conclusion in Section 7.

2. **Related Works.**

Data mining is a method of extracting non-trivial, previously unknown information. It has received much interest from researchers due to its extensive applications. Several data mining techniques have been developed. Clustering is one important technique that is used to find the dense as well as sparse regions in the dataset. There are two primary approaches to clustering: partitioning and hierarchical. In the partitioning approach, data instances are divided into a predefined number of clusters based on certain criteria.

Clustering algorithms for numeric data are discussed in [2], where the distance function is used as a criterion. In [15], the authors proposed an algorithm for anomaly detections using a concept of minimizing the compactness and maximizing the separation of the clusters, which in turn improves the quality of anomaly detection. In [16], the authors used clustering techniques for detecting anomaly in time-series data. A method of anomaly detection for log files is discussed in [17]. In [18], a system of anomaly detection in network connection logs is proposed; this technique investigates outliers using clustering, highlighting the subtle variances in each model through visualization. The proposed approach can be extrapolated to a more generalized system of analyzing connection logs across a large infrastructure containing a huge number of nodes.

Intrusion is an activity that can compromise the integrity, confidentiality, or availability of resources by accessing them in an illegitimate way. Intrusion activities and the methods for their detection are discussed in [9,10], where the fuzzy c-means algorithm is used for intrusion detections. A method based on the k-means algorithm for traffic anomaly detection using weighted Euclidean distance is discussed in [14]. In [19], the authors introduced a multi-stage intrusion analysis system known as traffic-log combined detection (TLCD), which can discover not only the steps of the cyber-attack process, but also the behaviour of normal users. In [20], the authors discussed the meaning of statistical outlier detection, database-related data mining methods for outlier detection, reviewed approaches to find a statistically meaningful interpretation of the outlier scores, and sketched related current research topics. In [21], the authors introduced the rough set theory in the intrusion detection system which is helpful in the identification of potentially malicious content.

In [22], the authors introduced a deep learning approach that improves the accuracy of vehicle intrusion detection systems. In [23], a hybrid intrusion detection alert system framework, which uses a distributed deep learning model for handling and analyzing large-scale data in real time, is discussed. In [24], the authors proposed a method called 'delayed short-term memory' for time-series data. This method is based on prediction errors, and it provides multiple models with delayed prediction. In [25], the authors presented a deep learning-based approach for the detection of anomalies in time-series data. The advantage to the method used in [25] is that it is capable of handling minor data contamination and accurate in detecting small deviations or anomalies in time-series cycles, which is overlooked by other well-known distance-based and density-based anomaly detection techniques. A method for finding alert correlations based on frequent itemset mining is discussed in [26], while classification-based intrusion detection system is discussed in [27]. In [28], the authors have presented an improved version of k-means algorithm which overcomes the drawback of k-means algorithm by optimally determining initial set of clusters.

3. Problem Definition.

In this section, we discuss some important definitions, notations and formulae used in the proposed algorithm. Since most real-life datasets are hybrids, and the k-means algorithm uses the distance between the object and the cluster, typical distance formulae do not work. Therefore, we need to develop a general distance formula that can be applicable to numeric, categorical, or hybrid attributes. The formulae are given below.

3.1. Distance in categorical attributes.

In [4], the authors proposed a distance formula for categorical attribute as follows. Let $A_1, A_2, \ldots, A_{d_c}$ be categorical attributes of a set of data instances which also have numeric attributes. The domain $(A_i; i = 1, 2, \ldots, d_c) = \{a_{i1}, a_{i2}, \ldots, a_{im}\}$ comprises finite, unordered possible values that can be taken by each attribute A_i, such that for any a, $b \in \mathrm{dom}(A_i)$, either $a = b$ or $a \neq b$. Any data instance x_i is a vector $(x_{i1}, x_{i2}, \ldots, x_{id_c})^{/}$, where $x_{ip} \in \mathrm{dom}(A_p)$, $p = 1, 2, \ldots, d_c$. The distance

$d(x_i, C_j)$ between data instance x_i and cluster C_j, $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, k$ is given by

$$d(x_i, C_j) = \sum_{p=1}^{d_c} w_p d(x_{ip}, C_j), \text{ with } \sum_{p=1}^{d_c} w_p + w_{d_c+1} = 1 \tag{1}$$

Here w_p is the weight factor associated with each categorical attribute and w_{d_c+1} that associated with all numeric part of the data instances respectively. The weights describe the importance of the attribute which controls the contribution attribute-cluster distance to data instance-cluster distance. The attribute-cluster distance between x_{ip} and C_j is given by

$$d(x_{ip}, C_j) = \frac{|C_j(A_p = x_{ip})|}{|C_j(A_p \neq \phi)|} \tag{2}$$

Obviously $d(x_{ip}, C_j) \in [0, 1]$ means $d(x_{ip}, C_j) = 1$ only if all the data instance in C_j has $A_p = x_{ip}$ and $d(x_{ip}, C_j) = 0$ only if no data instance in C_j has $A_p = x_{ip}$.

With the help of Equation (2), Equation (1) becomes

$$d(x_i, C_j) = \sum_{p=1}^{d_c} w_p d(x_{ip}, C_j) = \sum_{p=1}^{d_c} w_p \frac{|C_j(A_p = x_{ip})|}{|C_j(A_p \neq \phi)|} \tag{3}$$

where $d(x_i, C_j) \in [0, 1]$, $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, k$.

3.2. **Calculating the weight of an attribute.** The proposed method in [4,29] for calculating the weights of attributes is given below.

Let A be an attribute, and then, its importance I_A can be quantified by the following entropy metric.

$$I_A = -\int \rho(x(A)) \log(\rho(x(A)))dx(x) \tag{4}$$

where $x(A)$ is a value of the attribute A, and $\rho(x(A))$ is the distribution function of data in the cluster along the A dimension. Since the categorical attribute values are discrete and independent, the probability of an attribute can be estimated by counting the frequency of the attribute value. Consequently, the importance of any categorical attribute A_p ($p \in \{1, 2, \ldots, d_c\}$) can be calculated by the formula

$$I_{A_p} = -\sum_{t=1}^{m_p} \rho(a_{pt}) \log \rho(a_{pt}) \tag{5}$$

Furthermore,

$$\rho(a_{pt}) = \frac{|X(A_p = a_{pt})|}{|X(A_p \neq \phi)|}$$

where $a_{pt} \in \text{dom}(A_p)$, m_p is the number of possible values that A_p can take, and X is the whole dataset. Equation (5) states that the importance of an attribute is directly proportional to the number of different values of the categorical attribute. However, in practice, an attribute with vastly different values will contribute minimally to the cluster. For example, the ID number of an instance is not useful for cluster analysis. Hence, Equation (5) can further be modified as

$$I_{A_p} = -\frac{1}{m_p} \sum_{t=1}^{m_p} \rho(a_{pt}) \log \rho(a_{pt}) \tag{6}$$

Hence, the importance of an attribute can be qualified by its average entropy over each attribute value; thus the weight of each attribute can be computed as

$$w_p = \frac{I_{A_p}}{\sum_{t=1}^{d} I_{A_t}}, \quad p = 1, 2, \ldots, d_c \tag{7}$$

3.3. **Distance in numeric attributes.** The distance formula in [4] for numeric attributes of the data instance is defined as follows. Let $x_i = (x_{i1}, x_{i2}, \ldots, x_{id_n})$ be the numeric attribute of a data instance x_i, and then, the distance $d(x_i, C_j)$ between numeric attribute x_i, $i = 1, 2, \ldots, n$ and cluster C_j, $j = 1, 2, \ldots, k$ is defined as follows:

$$d(x_i, C_j) = \frac{e^{\left(-0.5||x_i - c_j||^2\right)}}{\sum_{t=1}^{k} e^{\left(-0.5||x_i - c_t||^2\right)}} \tag{8}$$

where c_j is the centroid of cluster C_j, and $d(x_i, C_j) \in [0, 1]$.

3.4. **Distance in mixed attributes.** Suppose the data instance $x_i = [x_i^c, x_i^n]$, where $x_i^c = \left(x_{i1}^c, x_{i2}^c, \ldots, x_{id_c}^c\right)$ and $x_i^n = (x_{i1}^n, x_{i2}^n, \ldots, x_{idn}^n)$ are categorical and numerical attributes of $x_i (d_c + d_n = d)$, respectively. Using Equations (1) and (8), the distance $d_1(x_i, C_j)$ between the data instance x_i and cluster C_j is defined [4] as follows:

$$d_1(x_i, C_j) = \sum_{p=1}^{d_c} w_p \frac{\left|C_j\left(A_p = x_{ip}^c\right)\right|}{|C_j(A_p \neq \phi)|} + w_{d_c+1} \frac{e^{\left(-0.5\left\|x_i^n - c_j^n\right\|^2\right)}}{\sum_{t=1}^{k} e^{\left(-0.5\left\|x_i^n - c_t^n\right\|^2\right)}} \tag{9}$$

Here, $\sum_{p=1}^{d_c+1} w_p = 1$ and c_j^n is the centroid of cluster C_j.

Since we compare the distances of the data instances with clusters and take the data instance with the minimum distance value to be in the cluster, we can rewrite $d(x_i, C_j)$ as follows:

$$d(x_i, C_j) = \left(1 - \sum_{p=1}^{d_c} w_p \frac{\left|C_j\left(A_p = x_{ip}^c\right)\right|}{|C_j(A_p \neq \phi)|}\right) + w_{d_c+1} \frac{e^{\left(-0.5\left\|x_i^n - c_j^n\right\|^2\right)}}{\sum_{t=1}^{k} e^{\left(-0.5\left\|x_i^n - c_t^n\right\|^2\right)}} \tag{10}$$

It is worth mentioning here that we subtract the distance in categorical attributes from 1 to fit it onto the same scale as the distance in numeric attributes. Obviously, $d(x_i, C_j) \in [0, 1]$. If $x_i \in C_j$, $d(x_i, C_j) = 0$. In Equation (10), the numerical attributes are included as a whole in the Euclidean distance, hence, it can be treated as one of the indivisible components, and only one weight can be assigned to it. Thus, we will have d_c+1 attribute weights in total, and their summation is equal to 1. Under these conditions, we can place the attribute weights at: $w_{d_c+1} = \frac{1}{d_c+1}$.

In addition,

$$w_p = \frac{d_c I_{A_p}}{(d_c + 1)\sum_{t=1}^{d} I_{A_p}}, \quad p = 1, 2, \ldots, d_c \tag{11}$$

Thus, the total weights of the numeric and categorical parts are $1/(d_c+1)$ and $d_c/(d_c+1)$, respectively. Moreover, as the actual weight of each categorical attribute is adjusted by its importance as in Equation (7), Equation (11) can give us the weights for mixed attributes. Obviously, $d(x_i, C_j) \in [0, 1]$.

3.5. **Similarity of the cluster pair.** Let C_i and C_j be two clusters obtained at any stage, where $i, j \in \{1, 2, \ldots, k\}$ and $i \neq j$, c_i = centroid of C_i and c_j = centroid of C_j then we define the similarity measure $S(C_i, C_j)$ between C_i and C_j as follows:

$$S(C_i, C_j) = (S_n(C_i, C_j) + 1 - S_c(C_i, C_j))/2 \tag{12}$$

where $S_n(C_i, C_j)$ is the similarity of C_i and C_j on numeric attributes

$$= w_{d_c+1} \frac{e^{-0.5\|c_i - c_j\|^2}}{\sum_{t=1}^{k} e^{-0.5\|c_i - c_t\|^2} + \sum_{t=1}^{k} e^{-0.5\|c_t - c_j\|^2}} \tag{13}$$

and $S_c(C_i, C_j)$ = the similarity of C_i and C_j on categorical attributes

$$= \sum_{p=1}^{d_c} w_p \frac{|C_i(A_p = x_{tp})| + |C_j(A_p = x_{tp})|}{|C_i(A_p)| + |C_j(A_p)|}; \quad t = 1, 2, \ldots, m \tag{14}$$

Using Equations (13) and (14), Equation (12) becomes

$$S(C_i, C_j) = \frac{w_{d_c+1} \frac{e^{-0.5\left\|c_i - c_j\right\|^2}}{\sum_{t=1}^{k} e^{-0.5\|c_i - c_t\|^2} + \sum_{t=1}^{k} e^{-0.5\left\|c_t - c_j\right\|^2}} + \left(1 - \sum_{p=1}^{d_c} w_p \frac{|C_i(A_p = x_{tp})| + |C_j(A_p = x_{tp})|}{|C_i(A_p \neq \phi)| + |C_j(A_p \neq \phi)|}\right)}{2} \quad (15)$$

In Equation (15), we subtract the similarity in categorical attributes from 1 to measure onto the same scale as that of numeric attributes. Since $S_n(C_i, C_j) \in [0, 1]$ and $S_c(C_i, C_j) \in [0, 1]$, it follows that $S(C_i, C_j) \in [0, 1]$. For identical cluster pairs, $S(C_i, C_j) = 0$, and $S(C_i, C_j) = 1$ for completely dissimilar pairs.

3.6. **Merge function.** Let C_i and C_j be the two clusters and C be the cluster formed by merging them, then, the merge() function [8,9] is defined as $C = merge(C_i, C_j) = C_i \cup C_j$, if and only if $S(C_i, C_j) \leq \sigma$, a pre-defined threshold.

4. **Proposed Algorithm.** It is already mentioned in Section 1 that the proposed algorithm uses the combination of both partitioning and hierarchical approaches and is a modified version of k-means where at the end of every iteration, a merge function is used on highly similar clusters which minimizes the number of output clusters. The algorithm is described as follows. First of all, the algorithm selects k-cluster centroids randomly from a given n input d-dimensional data instances. Next, we compute the distance function for each data instance x_i with each cluster C_j, $j = 1, 2, \ldots, k$, and put the data instance in the cluster with the minimum distance value. It is to be mentioned here that the weights of categorical attributes are taken to be equal. Subsequently, we update the frequency of categorical value and the centroid of the numeric value for each new cluster. In order to conveniently update the centroid of the clusters and the categorical attribute values, we maintain two auxiliary matrices for each cluster. One matrix is required to store the frequency of each categorical value occurring in the cluster, and the other stores the mean vector of the numerical parts of all the objects belonging to the cluster. After this we compute the weights of categorical attributes. Then we apply the merge function (merge function is defined in Section 3) to the set clusters obtained in previous step which will produce a smaller number of clusters of bigger size by merging the similar clusters. The process would continue until any of the following conditions are satisfied: (i) no changes to the data instances clusters and (ii) no merging of the clusters is possible. The algorithm is better described with a flowchart in Figure 1.

The pseudo code of the algorithm is given in Algorithm 1.

The algorithm supplies the cluster's data instances as well as the outliers – i.e., the data instances belonging to smaller clusters. The outliers can be used to identify anomalies among the data instances, and such patterns obtained by the algorithm can be used in designing an efficient intrusion detection system (IDS).

5. **Time Complexity.** To calculate the centroid of randomly selected k-data instances, the computation cost required is $O(n + nkd_n) = O(nkd_n)$, where n = the number of data instances, k = the number of clusters and d_n = the number of numerical attributes. Therefore, the total cost of steps 1 and 2 is $O(nkd_n)$. Step 3 takes $O(kn)$ time because, for each centroid, we have to compute the distance of each data instance, choose the minimum, and assign it to that cluster. We need $O(n)$ time to compute the minimum for each of the k clusters; thus the computation cost is $O(kn)$. In step 4, the cost of updating two auxiliary matrices is $O(2k)$. Also, the computation cost for calculating the weights of categorical attributes is $O(mnkd_c)$, where m = the average number of possible values that the categorical attributes can take, and d_c = the number of categorical attributes. Thus, the total computation cost of steps 4 and 5 is $O(2k + mnkd_c) = O(mnkd_c)$. The

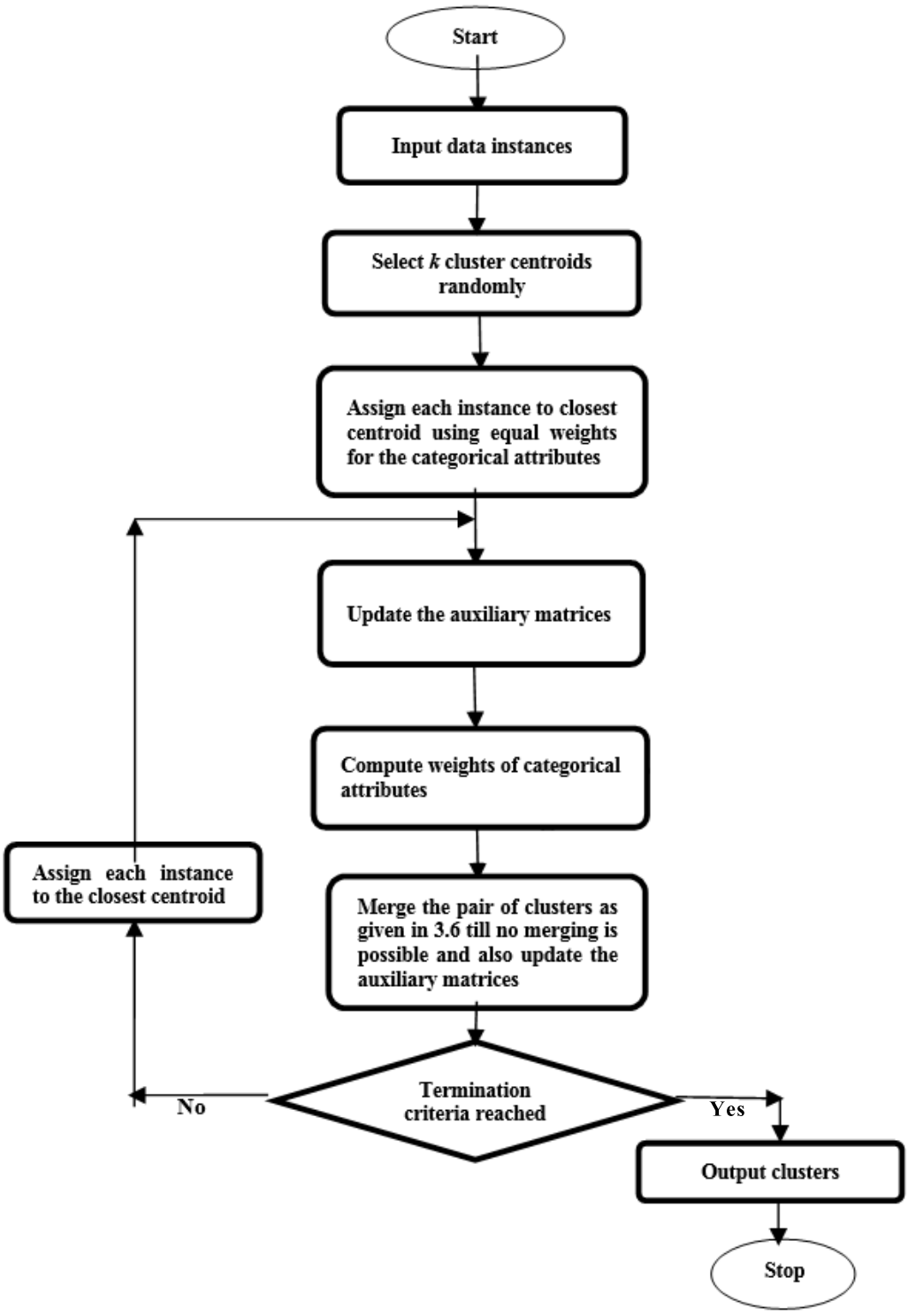

FIGURE 1. Flowchart of the proposed modified k-means algorithm

computation cost of step 6 is $\mathrm{O}(n_1 n_2) = \mathrm{O}\left(n^2\right)$, where n_1 and n_2 are the sizes of the two clusters to be merged and $n_1 \leq n$, $n_2 \leq n$. If the termination conditions are not satisfied then the cost of assigning each instance to the closest cluster is again $\mathrm{O}(kn)$. Thus, the total computation cost of one iteration is $\mathrm{O}\left(nkd_u + mnkd_c + n^2 + 2kn\right)$. If t is the number of iterations of the algorithm, then the total computation cost of the algorithm is $\mathrm{O}\left(t\left(nkd_u + mnkd_c + n^2 + 2kn\right)\right) = \mathrm{O}(t(mnkd))$, where $d = d_n + d_c$. Note that $t \leq k \leq n$, $m \leq n$, and $d \leq n$. Also k is constant, so t can also be treated as constant. Therefore, the overall complexity of the algorithm is $\mathrm{O}(t(mnkd)) = \mathrm{O}\left(n^3\right)$, and the proposed algorithm is quite efficient.

Algorithm 1. Modified *k*-means algorithm for anomaly detection.

Step 1. Given n, d-dimensional data instances as input
Step 2. Select k-cluster centroids randomly.
Step 3. Assign each instance to the closest centroid using equal weights for the categorical attributes.
Step 4. Update the two auxiliary matrices maintained for storing the frequency of each categorical value occurring in the cluster, and the mean vector of the numerical parts of all the data instances belonging to the cluster.
Step 5. Compute the weights of categorical attributes.
Step 6. Merge the pair of clusters using similarity measure given in (15)
Step 7. If ((no changes to the data instances clusters) // (no merging of the clusters is possible))
  Output clusters
  else
  assign each instance to the closest centroid
  go to step 4.

6. **Experimental Results.** For experiment, we take two different datasets [30] viz. Flags Dataset and KDD99 Dataset. Flags Dataset contains details of various countries and flags with 10 numeric-valued and 30 categorical-valued attributes, whereas KDD99 Dataset is one of the popular datasets used for building network intrusion detector, a predictive model to identify intrusions or attacks, from normal connections. A summarized view of the dataset describing the dataset characteristics, the attribute characteristics, the number of attributes, and the number of data instances is presented in Table 1.

Table 1. Datasets characteristics

Datasets	Dataset char[c]	Attribute char[c]	Number of attributes (numeric/categorical)	No of data instances
Flags Dataset	Multivariate	Numeric, Boolean or nominal	10/30	209
KDD99 Dataset	Multivariate	Categorical, Integer	34/8	4898431

The experiments were conducted using Matlab on Intel Core i7-2600 machine with 3.4 GHz, 8 M Cache, 8 GB RAM, 500 GB Hard disc running Windows 10. The results were reported in Table 2 both as an absolute anomaly and anomaly ratio. Thus the obtained results further confirm the efficacy of our algorithm.

Table 2. Obtained results

Datasets	No of anomalies	Anomaly ratio (no of anomalies)/(no of data instances)
Flags Dataset	2	0.009569
KDD99 Dataset	160	0.00003276

7. **Conclusion.** The anomaly detection technique is one method for developing intrusion detection systems. In this paper, we proposed an algorithm which uses both partitioning and hierarchical approaches. The algorithm is used for detecting anomaly in network data. For this purpose, we have suitably defined the distance, similarity, and merge functions, which work on both numeric and categorical attributes. The algorithm supplies the number of clusters less than or equal to a pre-defined value, with some outliers – i.e., data instances belonging to smaller clusters. The extracted outliers, which deviate from the

normal behaviour of the data, are considered anomalies. Such anomalies can be used in designing an efficient intrusion detection system.

Although our proposed algorithm looks like well-known k-means algorithm, it differs from the aforesaid algorithm in the following points.

1) The k-means algorithm is successfully used for numeric data, however our algorithm can be applied for numeric, categorical or mixture of both because the distance function is redefined in such a way that it can work for above-mentioned datasets.

2) Secondly, the output obtained by k-means algorithm depends heavily on the initial parameters like initial set of clusters and minimum threshold; however, our algorithm is less dependent on the initial parameters.

3) At the end every iteration, similar cluster pairs are merged using a suitably defined similarity measure which in turn reduces the number of clusters and hence decreases the number of executions of algorithm.

REFERENCES

[1] K. Bailey, *Numerical Taxonomy and Cluster Analysis*, Typologies and Taxonomies, 1994.

[2] J. A. Hartigan, *Clustering Algorithms*, John Wiley & Sons, 1975.

[3] D. Gibson, J. Kleinberg and P. Raghavan, Clustering categorical data: An approach based on dynamical system, *Proc. of the 24th Int'l Conf. on Very Large Databases*, New York, pp.311-323, 1998.

[4] Y.-M. Cheng and H. Jia, *A Unified Metric for Categorical and Numeric Attributes in Data Clustering*, Hong Kong University Technical Report, http://www.comp.hkbu.edu.hk/tech-report, 2011.

[5] F. A. Mazarbhuiya and M. Abulaish, Clustering periodic patterns using fuzzy statistical parameters, *International Journal of Innovative Computing, Information and Control*, vol.8, no.3(b), pp.2113-2124, 2012.

[6] F. A. Mazarbhuiya, M. Y. AlZahrani and L. Georgieva, Anomaly detection using agglomerative hierarchical clustering algorithm, in *Information Science and Applications 2018. ICISA 2018. Lecture Notes in Electrical Engineering*, K. Kim and N. Baek (eds.), Singapore, Springer, DOI: 10.1007/978-981-13-1056-0_48, 2018.

[7] R. Pamula, J. K. Deka and S. Nandi, An outlier detection method based on clustering, *Proc. of the 2011 2nd International Conference on Emerging Applications of Information Technology*, pp.253-256, 2011.

[8] Y. Zhang, J. Liu and H. Li, An outlier detection algorithm based on clustering analysis, *Proc. of the 2010 1st International Conference on Pervasive Computing, Signal Processing and Applications*, 2010.

[9] D. Sharma, Fuzzy clustering as an intrusion detection technique, *International Journal of Computer Science & Communication Networks*, vol.1, no.1, pp.69-75, 2011.

[10] L. Xie, Y. Wang, L. Chen and G. Yue, An anomaly detection method based on fuzzy c-means clustering algorithms, *Proc. of the 2nd Symposium on Networking and Network Security*, China, pp.89-92, 2010.

[11] H. Debar, M. Dacier and A. Wespi, Towards a taxonomy of intrusion detection systems, *Computer Networks*, vol.31, pp.805-822, 1999.

[12] T. Escamilla, *Intrusion Detection: Network Security beyond the Firewall*, John Wiley & Sons, New York, 1998.

[13] G. Munz, S. Li and G. Carle, *Traffic Anomaly Detection Using K-Means Clustering*, Allen Institute for Artificial Intelligence, 2007.

[14] W. Ren, J. Cao and X. Wu, Application of network intrusion detection based on fuzzy c-means clustering algorithm, *The 3rd International Symposium on Intelligent Information Technology Application*, pp.19-22, 2009.

[15] R. Alguliyev, R. Aliguliyev and L. Sukhostat, Anomaly detection in big data based on clustering, *Statistics, Optimization and Information Computing*, vol.5, pp.325-340, 2017.

[16] M. Landauer, M. Wurzenberger, F. Skopik, G. Settani and P. Filzmoser, Time-series analysis: Unsupervised anomaly detection beyond outlier detection, *Proc. of the International Conference on Information Security, Practice and Experience (ISPEC-2018)*, 2018.

[17] M. Landauer, M. Wurzenberger, F. Skopik, G. Settani and P. Filzmoser, Dynamic log file analysis: An unsupervised cluster evaluation approach for anomaly detection, *Computer & Security*, vol.79, pp.94-116, 2018.

[18] S. Mehta, P. Kothuri and D. L. Garcia, Anomaly detection for network connection logs, *arXiv Preprint*, arXiv:1812.01941, 2018.
[19] J. Lu, F. Lv, Z. Zhuo, X. Zhang, X. Liu, T. Hu and W. Deng, Integrating traffics with network device logs for anomaly detection, *Security and Communication Network, Big Data Analytics for Cyber Security*, vol.205, 2019.
[20] A. Zimek and P. Filmoser, There and back again: Outlier detection between statistical reasoning and data mining algorithms, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol.8, no.6, 2018.
[21] N. Gupta, R. Prasad, P. Saurabh and B. Verma, NB tree-based intrusion detection technique using rough set theory model, *Data Engineering and Applications*, pp.93-101, 2019.
[22] J. Zhang, F. Li, H. Zhang, R. Li and Y. Li, Intrusion detection system using deep learning for in-vehicle security, *Ad Hoc Network*, vol.95, 2019.
[23] R. Vijaykumar, M. Alazab, K. P. Somen, P. Poornachandran and A. Al-Memrat, Deep learning approach for intelligent intrusion detection system, *IEEE Access*, vol.7, pp.41525-41550, 2019.
[24] S. Maya, K. Ueno and T. Nishikawa, dLSTM: A new approach for anomaly detection using learning with delayed prediction, *International Journal of Data Science and Analytics*, vol.8, no.2, pp.137-164, 2019.
[25] M. Munir and S. A. Siddiqui, DeepAnT: A deep learning approach for unsupervised anomaly detection in time-series, *IEEE Access*, vol.7, pp.1991-2005, 2019.
[26] X. Feng, D. Wang, M. Huang and X. Sun, An approach of discovering causal knowledge for alert correlating based on data mining, *Proc. of the 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing (DASC)*, pp.57-62, 2014.
[27] R. Wankhede and V. Chole, Intrusion detection system using classification technique, *International Journal of Computer Applications*, vol.139, no.11, pp.25-28, 2016.
[28] Z. Khan, J. Ni, X. Fan and P. Shi, An improved K-means clustering algorithm based on an adaptive initial parameter estimation procedure for image segmentation, *International Journal of Innovative Computing, Information and Control*, vol.13, no.5, pp.1509-1525, 2017.
[29] J. Basak and R. Krishnapuram, Interpretable hierarchical clustering by constructing and unsupervised decision tree, *IEEE Trans. Knowledge and Data Engineering*, vol.17, no.1, pp.121-132, 2005.
[30] M. Lichman, *UCI Machine Learning Repository*, http://archive.ics.uci.edu/mll, 2013.