STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **5**, December 2017, pp 325–340.
Published online in International Academic Press (www.IAPress.org)

Anomaly detection in Big data based on clustering

Rasim Alguliyev, Ramiz Aliguliyev, Lyudmila Sukhostat [*]

Institute of Information Technology, Azerbaijan National Academy of Sciences, Baku, Azerbaijan.

(Received: 26 August 2017; Accepted: 22 October 2017)

Abstract Selection of the right tool for anomaly (outlier) detection in Big data is an urgent task. In this paper algorithms for data clustering and outlier detection that take into account the compactness and separation of clusters are provided. We consider the features of their use in this capacity. Numerical experiments on real data of different sizes demonstrate the effectiveness of the proposed algorithms.

Keywords Outlier Detection, Anomaly Detection, Big Data, Clustering, Clusters' Compactness, Clusters' Separation

AMS 2010 subject classifications 62H30, 62-07

DOI: 10.19139/soic.v5i4.365

1. Introduction

When analyzing data, the information quality is of paramount importance. This task is complicated by the growth of large volumes of collected information. Working with Big data requires large computational resources. In this regard, researchers pay special attention to the development of effective methods for anomaly (outlier) detection.

The high degree of importance of the solving tasks had led to the fact that a whole galaxy of different methods appeared in this area. The methods differ from each other in ease of implementation, suitability for data processing, and the basic principles underlying them.

Among of them are clustering methods. Clustering technology is used in many areas: medicine, archeology, information systems, etc. Data clustering is also often used as an initial step for data analytics.

The aim of this paper is to develop a clustering approach for anomaly detection in real Big data. The paper develops algorithms that minimize the compactness of clusters and maximize the separation of clusters from each other according to the distances between their centers and the remoteness of cluster centers from the selected common center of points in dataset.

The number of clusters is not known in advance and is established in accordance with some subjective criterion. Therefore, in this paper, the number of clusters is determined according to [1].

To illustrate the viability of the developed anomaly detection algorithms, the results of examples of small, medium and large real data sets are presented.

The rest of the paper is organized as follows. Section 2 gives a literature review of existing works on clustering of large amounts of data and outlier detection. The proposed clustering algorithms are described in Section 3. In Section 4, datasets and clustering evaluation metrics are presented. The experimental results and discussion are given in Section 5, followed by conclusions in Section 6.

[*] Correspondence to: Lyudmila Sukhostat (Email: lsuhostat@hotmail.com). Institute of Information Technology, Azerbaijan National Academy of Sciences. 9A, B. Vahabzade Street, Baku AZ1141, Azerbaijan.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

2. Related Work

Jiang et al. [2] presented two different initialization algorithms for k-modes clustering: 1) based on the traditional distance-based outlier detection technique; 2) based on the partition entropy-based outlier detection technique. The selection of initial cluster centers in k-modes clustering with the detection of outliers was combined. A weighted matching distance metric was adopted to calculate the distance between two objects described by categorical attributes.

In [3], an iterative procedure of clustering method based on multivariate outlier detection was proposed by using the Mahalanobis distance. At each iteration, a multivariate test of mean used to check the discrimination between the outlier clusters and the inliers. Multivariate control charts were used to graphically visualize the iterations and outlier clustering process.

A computationally efficient algorithm for the outlier detection in large volumes of information based on Rough Set Theory was presented by Macia-Perez et al. [4]. The proposed algorithm is applicable to both continuous and discrete data. It allows to locate (within specified times) elements in a data set that differ from the rest in some degree, errors in data collection to discard or correct, elements of a system having a malfunction, etc.

In [5], a hierarchical k-means (H-K-means) method for better clustering performance for Big data problems was proposed. This method simplifies the dataset, and then restores it back to the original one gradually with the succession of high-quality initial centroids. The proposed method is applied to a large-scale AMI (Advanced Metering Infrastructure) dataset and its effectiveness is evaluated by benchmarking with several existing clustering methods in terms of common adequacy indices, outlier detection, and computation time.

Liu et al. [6] presented a novel outlier detection approach to address data with imperfect labels and incorporate limited abnormal examples into learning. To deal with such data, the authors introduced likelihood values for each input data which denote the degree of membership of an example toward the normal and abnormal classes respectively. The proposed approach integrates local and global outlier detection and enhances the performance of outlier detection.

Souza and Amazonas [7] have proposed an outlier detection procedure using the k-means algorithm and Big data processing using the Hadoop platform and Mahout implementation integrated with Internet of Things architecture. The raw data was processed only once in the middleware layer, so different applications may be simpler. The proposed algorithm makes the application receive all instances without outliers and eliminates the overhead to analyse the raw data.

A new density-based algorithm for outlier detection was proposed in [8]. Natural Neighbor method was used to adaptively obtain the parameter, named Natural Value. To measure the outliers Natural Outlier Factor (NOF) was considered. In addition, Natural Value can be used in other outlier detection algorithms such as LOF (local outlier factor) [9] and INS (outlier detection algorithm using the instability factor [10] to achieve good results.

In [11], an efficient approximation to the k-means problem intended for Big data was proposed. The approach recursively partitions the entire dataset into a small number of subsets, each of which is characterized by its center of mass and weight, which can reduce the number of computed distances. It outperforms the K-means++ and the mini-batch K-means methods.

A novel approach based on clustering and outlier detection formulated as an integer program was presented in [12]. The proposed optimization problem enforces valid clusters as well as the selection of a fixed number of outliers. The modifications of the method based on Lagrangian duality were described to process large scale datasets.

In [13] the KMOR (k-means with outlier removal) algorithm motivated by work [14] was proposed. The idea of the algorithm is the introduction of an additional ”cluster” that contains all outliers. The KMOR algorithm assigns all outliers into a group naturally during the clustering process. The results of the experiments have shown that the KMOR algorithm is able to cluster data and detect outliers simultaneously and is able to outperform other algorithms in terms of accuracy and runtime.

3. Proposed Algorithms

This section describes the proposed algorithms for anomaly detection.

Let us denote the following notations: $x_i \in R^n (i = \overline{1,n})$ is the point from the dataset, where n is the total number of points in the input dataset, $c_p \in R^k (p = \overline{1,k})$ is the cluster's number, where k is the number of clusters, S_W is the compactness of clusters, S_{BW} is the separation of clusters from each other, and S_B is the measure of the remoteness of each cluster center (O_p) from the center of all points (O) in the input dataset

$$S_W = \sum_{p=1}^{k} \sum_{i=1}^{n} (x_i - O_p)(x_i - O_p)^T , \tag{1}$$

$$S_{BW} = \sum_{p=1}^{k-1} \sum_{q=p+1}^{k} (O_p - O_q)(O_p - O_q)^T , \tag{2}$$

$$S_B = \sum_{p=1}^{k} (O_p - O)(O - O_p)^T , \tag{3}$$

$$O = \frac{1}{n} \sum_{i=1}^{n} x_i, \ O_p = \frac{1}{n_p} \sum_{x_i \in C_p} x_i, \ n_p = |C_p| , \ p = 1, 2, ..., k. \tag{4}$$

The algorithm of the first proposed approach for anomaly detection is as follows:

Algorithm 1
Input: $X = \{x_1, x_2, ..., x_n\}$,
k: a number of clusters.

Output: Vector of cluster indices $IDX = \{idx_1, idx_2, ..., idx_n\}$.

Step 1: Find the center of all points of the dataset (O)

Step 2: $s = 0$

Step 3: Calculate the compactness (S_W) according to (1)

Step 4: Calculate the separation of clusters (S_{BW}) according to (2)

Step 5: Calculate the remoteness (S_B) according to (3)

Step 6: Calculate the value of the following function taking into account (1)-(4)

$$F_1^{(s)}(x) = \frac{S_B^{(s)} + S_{BW}^{(s)}}{S_W^{(s)}} \to max \tag{5}$$

Step 7: $s = s + 1$

Step 8: Repeat steps 3-7 until the convergence condition is met:

$$\left| \frac{f^{(s+1)} - f^{(s)}}{f^{(s)}} \right| \leqslant \varepsilon, \tag{6}$$

where s is the number of iteration steps.

Step 9: Return the values of IDX

End

In the second algorithm, the task is to maximize an objective function in order to detect anomalies in the dataset:

```
Algorithm 2
Input: $X = \{x_1, x_2, ..., x_n\}$,
        $k$: a number of clusters.
Output: Vector of cluster indices $IDX = \{idx_1, idx_2, ..., idx_n\}$.
Step 1: Find the center of all points of the dataset $(O)$
Step 2: $s = 0$
Step 3: Calculate the compactness $(S_W)$ according to (1)
Step 4: Calculate the separation of clusters $(S_{BW})$ according to (2)
Step 5: Calculate the remoteness $(S_B)$ according to (3)
Step 6: Calculate the value of the following function taking into account (1)-(4)

        $$F_2^{(s)}(x) = \frac{S_B^{(s)} * S_{BW}^{(s)}}{S_W^{(s)}} \to max \tag{7}$$

Step 7: $s = s + 1$
Step 8: Repeat steps 3-7 until the convergence condition is met:

        $$\left| \frac{f^{(s+1)} - f^{(s)}}{f^{(s)}} \right| \leqslant \varepsilon, \tag{8}$$

where $s$ is the number of iteration steps.
Step 9: Return the values of $IDX$
End
```

In the third algorithm, the task consists in maximizing the objective function according to regularization parameter α $(0 \leqslant \alpha \leqslant 1)$, which will be determined experimentally.
The steps of the algorithm are as follows:

```
Algorithm 3
Input: $X = \{x_1, x_2, ..., x_n\}$,
        $\alpha$: regularization parameter,
        $k$: a number of clusters.
Output: Vector of cluster indices $IDX = \{idx_1, idx_2, ..., idx_n\}$.
Step 1: Find the center of all points of the dataset $(O)$
Step 2: $s = 0$
Step 3: Calculate the compactness $(S_W)$ according to (1)
Step 4: Calculate the separation of clusters $(S_{BW})$ according to (2)
Step 5: Calculate the remoteness $(S_B)$ according to (3)
Step 6: Calculate the value of the following function taking into account (1)-(4)

        $$F_3^{(s)}(x) = \frac{1}{S_W^{(s)}} \left( \alpha S_B^{(s)} + (1 - \alpha) S_{BW}^{(s)} \right) \to max \tag{9}$$

Step 7: $s = s + 1$
Step 8: Repeat steps 3-7 until the convergence condition is met:

        $$\left| \frac{f^{(s+1)} - f^{(s)}}{f^{(s)}} \right| \leqslant \varepsilon, \tag{10}$$

where $s$ is the number of iteration steps.
Step 9: Return the values of $IDX$
End
```

4. Datasets and Evaluation Metrics

This section describes the datasets that were used to conduct the experiments and evaluation metrics.

4.1. Datasets

The experiments were performed on six datasets from the UCI repository [15, 16], including Diabetic Retinopathy Debrecen dataset (Diabetic), Phishing dataset, Banknote authentication, Forest CoverType dataset (Covertype), NSL-KDD dataset [17] and Spambase dataset (Spam).

Diabetic Retinopathy (DR) Debrecen Dataset contains features extracted from the Messidor image set to predict whether an image contains signs of diabetic retinopathy or not [18, 19]. It contains 19 features (a Euclidean distance of the center of the macula and the center of the optic disc, the binary result of the AM/FM-based classification, etc.) with 1151 samples.

Banknote Authentication Dataset was extracted from images that were taken from genuine and forged banknote-like specimens [16]. Dataset contains four features (variance of WT image, the skewness of WT image, the kurtosis of WT image, and entropy of image) with 1372 samples.

NSL-KDD Dataset of attack signatures was constructed based on KDD-99 database [20]. The database contains training (125973 samples) and test (22544 samples) sets. Labels are assigned to each instance either as an ”attack” type or as ”normal” behavior. The total number of samples (148517) was considered in this paper.

Covertype Dataset includes information about four wilderness areas located in the Roosevelt National Forest of northern Colorado (USA) [21]. Dataset classes include Spruce/Fir (1), Lodgepole Pine (2), Ponderosa Pine (3), Cottonwood/Willow (4), Aspen (5), Douglas-fir (6) and Krummholz (7). In this paper 1-6 classes were considered as normal values, and samples with class label Krummholz – as anomalies.

Spambase Dataset contains spam and non-spam e-mails [16] It includes features that indicate whether a particular word or character was frequently occurring in the e-mail and measure the length of sequences of consecutive capital letters. The dataset contains two classes: spam (1) or not (0).

Phishing Dataset contains 11055 phishing websites [22]. It includes 30 attributes (using the IP address, URL length, abnormal URL, website forwarding, etc.). This data belongs to one of the two classes labeled as Phishy (-1) and Legitimate (1).

4.2. Clustering Evaluation Metrics

Assume that the dataset N is divided into classes $C^+ = \left(C_1^+, ..., C_{k^+}^+\right)$ (true clustering), and, using the clustering procedure, clusters $C = (C_1, ..., C_k)$ can be found in the dataset [23].

A comparison of the clustering solutions is based on counting the pairs of points. Based on the results, a decision will be made: ”normal”/abnormal behavior. The most well-known clustering distance metrics based on data point pairs are the purity [24, 25], the Mirkin metric [26], the partition coefficient [27], the variation of information [28], the F-measure [29] and the V-measure [29].

Purity. The purity of the cluster C_p gives the ratio of the dominant class size in the cluster to the cluster size itself [24, 25, 30]. The value of the purity is always in the interval $\left[\frac{1}{k^+}, 1\right]$. The purity of the entire collection of clusters can be evaluated as a weighted sum of the individual cluster purities:

$$purity(C) = \frac{1}{n} \sum_{p=1}^{k} \max_{p^+=1,...,k^+} \left| C_p \bigcap C_{p^+}^+ \right|, \tag{11}$$

where k^+ is the initial number of classes, k is the number of clusters that need to be found. A higher purity value indicates a better clustering solution.

Mirkin metric. The Mirkin metric is defined as follows [26]:

$$M(C, C^+) = \frac{1}{n^2} \left(\sum_{p=1}^{k} |C_p|^2 + \sum_{p^+=1}^{k^+} \left| C_{p^+}^+ \right|^2 - 2 \sum_{p=1}^{k} \sum_{p^+=1}^{k^+} \left| C_p \bigcap C_{p^+}^+ \right|^2 \right). \tag{12}$$

The smaller the metric value, the better clustering.

F-measure. Another evaluation measure, also known as the clustering accuracy, is based on the F value of the cluster C_p and the class $C^+_{p^+}$, that is the harmonic mean of the precision and the recall. Precision and recall are computed as follows [23]:

$$P\left(C_p, C^+_{p^+}\right) = \frac{\left|C_p \bigcap C^+_{p^+}\right|}{|C_p|}, \tag{13}$$

$$R\left(C_p, C^+_{p^+}\right) = \frac{\left|C_p \bigcap C^+_{p^+}\right|}{\left|C^+_{p^+}\right|}. \tag{14}$$

Thus, the F-measure has the following form:

$$F\left(C_p, C^+_{p^+}\right) = \frac{2P\left(C_p, C^+_{p^+}\right) R\left(C_p, C^+_{p^+}\right)}{P\left(C_p, C^+_{p^+}\right) + R\left(C_p, C^+_{p^+}\right)}. \tag{15}$$

The F-measure of the cluster C_p is the maximum F-value attained at any class in the entire set of classes $C^+ = \left(C^+_1, ..., C^+_{k^+}\right)$. The F-measure of the entire dataset is considered to be the weighted sum of the individual cluster F-measures. That is,

$$F\left(C\right) = \sum_{p=1}^{k} \frac{|C_p|}{n} \max_{C^+_{p^+} \in C^+} F\left(C_p, C^+_{p^+}\right). \tag{16}$$

The higher the F-measure, the better clustering solution.

Partition coefficient (PC). This coefficient is used to compare $C = (C_1, ..., C_k)$ and $C^+ = \left(C^+_1, ..., C^+_{k^+}\right)$ distributions [27]. According to [31], PC is calculated as:

$$PC\left(C, C^+\right) = \frac{1}{kk^+} \sum_{p=1}^{k} \sum_{p^+=1}^{k^+} \left(\frac{\left|C_p \bigcap C^+_{p^+}\right|}{|C_p|}\right)^2. \tag{17}$$

A higher value of $PC\left(C, C^+\right)$ indicates a better clustering solution.

Variation of information (VI). This metric measures the amount of information that the authors gain and lose when going from the clustering C to another clustering C^+ [28, 31].

$$VI\left(C, C^+\right) = \frac{1}{n \log n} \sum_{p=1}^{k} \sum_{p^+=1}^{k^+} \left|C_p \bigcap C^+_{p^+}\right| \log\left(\frac{|C_p| \left|C^+_{p^+}\right|}{\left|C_p \bigcap C^+_{p^+}\right|^2}\right). \tag{18}$$

In general, the smaller the VI, the better clustering solution.

V-measure. The V-measure is an entropy-based measure that explicitly measures how successfully the criteria of homogeneity and completeness have been satisfied [25]. The homogeneity can be defined as

$$hom(C) = \begin{cases} 1, & if\ H\left(C^+|C\right) = 0 \\ 1 - \frac{H\left(C^+|C\right)}{H\left(C^+\right)}, & else \end{cases}, \tag{19}$$

where

$$H\left(C^+|C\right) = -\sum_{p=1}^{k} \sum_{p^+=1}^{k^+} \frac{\left|C_p \bigcap C^+_{p^+}\right|}{n} \log\left(\frac{\left|C_p \bigcap C^+_{p^+}\right|}{\sum_{p^+=1}^{k^+} \left|C_p \bigcap C^+_{p^+}\right|}\right), \tag{20}$$

$$H\left(C^{+}\right)=-\sum_{p^{+}=1}^{k^{+}}\frac{\sum_{p=1}^{k}\left|C_p\bigcap C_{p^{+}}^{+}\right|}{k^{+}}\log\left(\frac{\sum_{p=1}^{k}\left|C_p\bigcap C_{p^{+}}^{+}\right|}{k^{+}}\right). \tag{21}$$

$H\left(C^{+}|C\right)$ is equal to 0 when each cluster contains only members of a single class, a perfect homogeneous clustering. In the degenerate case when $H\left(C^{+}\right)$ is equal to 0, when there is only a single class, the homogeneity is defined to be 1.

Completeness is symmetric to homogeneity. The completeness can be defined as

$$comp(C)=\begin{cases}1, & if\ H\left(C|C^{+}\right)=0\\ 1-\frac{H\left(C|C^{+}\right)}{H(C)}, & else\end{cases}, \tag{22}$$

where

$$H\left(C|C^{+}\right)=-\sum_{p^{+}=1}^{k^{+}}\sum_{p=1}^{k}\frac{\left|C_p\bigcap C_{p^{+}}^{+}\right|}{n}\log\left(\frac{\left|C_p\bigcap C_{p^{+}}^{+}\right|}{\sum_{p=1}^{k}\left|C_p\bigcap C_{p^{+}}^{+}\right|}\right), \tag{23}$$

$$H\left(C\right)=-\sum_{p=1}^{k}\frac{\sum_{p^{+}=1}^{k^{+}}\left|C_p\bigcap C_{p^{+}}^{+}\right|}{k^{+}}\log\left(\frac{\sum_{p^{+}=1}^{k^{+}}\left|C_p\bigcap C_{p^{+}}^{+}\right|}{k^{+}}\right). \tag{24}$$

V-measure of the clustering solution is calculated by finding the harmonic mean of homogeneity and completeness as follows:

$$V(C)=\frac{2hom(C)comp(C)}{hom(C)+comp(C)}. \tag{25}$$

The computation of the homogeneity, the completeness, and the V-measure are completely independent from the number of classes and clusters, the size of the dataset and the clustering algorithm.

5. Experimental Results and Discussion

A number of experiments were implemented in Matlab 2016a on a 64-bit Windows-based system with an Intel core (i7), 2.5 GHz processor machine with 8 Gbytes of RAM to evaluate the performance of the proposed algorithms.

Experimental datasets Diabetic, Phishing, NSL-KDD, Banknote authentication, Spam, and Covertype were used as initial data. The characteristics of the datasets are presented in Table 1. Six quality metrics having different nature were selected for the analysis.

Table 1. Summary of the datasets.

Dataset	*Number of instances*	C_1^{+}	C_2^{+}	*Number of attributes*
Diabetic	1151	611	540	19
Phishing	11055	4898	6157	30
NSL-KDD	148517	71463	77054	41
Banknote authentication	1372	762	610	4
Spam	4601	2788	1813	57
Covertype	581012	20510	560502	54

The datasets were divided into two classes: C_1^{+} and C_2^{+}. During the preprocessing, the values in the datasets were standardized. Samples in the C_1^{+} class were taken as anomalies.

The results of the proposed algorithms based on six metrics are presented below. Purity, Mirkin metric, F-measure, PC, VI, and V-measure were considered as evaluation metrics. The best results were marked in bold.

The proposed approaches are compared with the k-means algorithm. Table 2 shows the experimental results on all datasets for the k-means algorithm.

Table 2. Performance evaluation of k-means algorithm on all datasets.

Dataset	C_1	C_2	*Purity*	*Mirkin*	*F-measure*	*VI*	*PC*	*V-measure*
Diabetic	859	292	0.5308	0.5000	0.5931	0.1783	0.2519	1.0003
Phishing	9263	1792	0.5569	0.4953	0.5888	0.1213	0.2523	1.0000
NSL-KDD	1834	146683	0.5188	0.4993	**0.6801**	0.0637	0.2502	1.0000
Banknote authentication	462	910	0.6122	0.4748	0.5637	0.1780	0.2615	1.0003
Spam	244	4357	0.6359	0.4630	0.5385	0.0999	0.2981	1.0000
Covertype	180133	400879	**0.9647**	**0.4315**	0.5859	**0.0580**	**0.4625**	1.0000

The best result according to Purity, Mirkin, VI, and PC metrics was obtained for the Covertype dataset and gained 96.47%, 43.15%, 5.8% and 46.25%, respectively. F-measure showed the best result on the NSL-KDD dataset. According to Purity (53.08%), Mirkin (50%) and VI (17.83%) metrics, the lowest value was achieved for the Diabetic dataset.

The best results for the first proposed algorithm were obtained for the Covertype dataset: Purity = 96.47%, Mirkin metric = 8.04%, F-measure = 97.23%, VI= 1.46%, PC = 47.55% and V-measure = 1.0000 (Table 3).

Table 3. Performance evaluation of the first proposed algorithm on different datasets.

Dataset	C_1	C_2	*Purity*	*Mirkin*	*F-measure*	*VI*	*PC*	*V-measure*
Diabetic	871	280	0.5752	0.4887	0.6291	0.1702	0.2742	1.0003
Phishing	601	10454	0.5717	0.4897	0.7081	0.0955	0.2615	1.0000
NSL-KDD	1415	147102	0.5226	0.4990	0.6833	0.0625	0.2700	1.0000
Banknote authentication	175	1197	0.6436	0.4588	0.5418	0.1340	0.3163	1.0001
Spam	252	4349	0.6242	0.4691	0.5462	0.1026	0.2713	1.0000
Covertype	3977	577035	**0.9647**	**0.0804**	**0.9723**	**0.0146**	**0.4755**	1.0000

The analysis reveals that the second proposed approach yields a high quality of clustering for NSL-KDD according to Mirkin and F-measure metrics and for Covertype dataset according to Purity, VI and PC metrics (Table 4). The worst results were also obtained for Diabetic dataset.

Table 4. Performance evaluation of the second proposed algorithm on different datasets.

Dataset	C_1	C_2	*Purity*	*Mirkin*	*F-measure*	*VI*	*PC*	*V-measure*
Diabetic	783	368	0.5804	0.4871	0.5984	0.1814	0.2658	1.0003
Phishing	3023	8032	0.6715	0.4412	0.6357	0.1122	0.3403	1.0000
NSL-KDD	85159	63358	0.8215	**0.2932**	**0.8176**	0.0774	0.3570	1.0000
Banknote authentication	976	396	0.7172	0.4056	0.7299	0.1518	0.3076	1.0002
Spam	4330	271	0.6405	0.4605	0.7531	0.1011	0.3016	1.0000
Covertype	400877	180135	**0.9647**	0.4315	0.3118	**0.0580**	**0.4625**	1.0000

The influence of the regularization parameter α on the performance of the proposed algorithm on different datasets was considered (Table 5-15). For α, the authors used the values 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 and 1.

It can be concluded from Table 5-8 that α=0, α=0.1, α=0.2 and α=0.3 give the best results for the Covertype dataset.

Table 5. Performance evaluation of the third proposed algorithm for $\alpha = 0$.

Dataset	C_1	C_2	*Purity*	*Mirkin*	*F-measure*	*VI*	*PC*	*V-measure*
Diabetic	1015	136	0.5621	0.4923	0.5965	0.1476	0.2601	1.0001
Phishing	705	10350	0.5952	0.4819	0.7187	0.0953	0.2983	1.0000
NSL-KDD	8186	140331	0.5305	0.4981	0.6394	0.0716	0.3503	1.00000
Banknote authentication	685	687	0.9437	0.1059	0.9437	0.0495	0.4502	1.0001
Spam	1820	2781	0.7003	0.4198	0.6726	0.1425	0.2892	1.0001
Covertype	10876	570136	**0.9647**	**0.1020**	**0.9554**	**0.0184**	**0.4815**	1.0000

Table 6. Performance evaluation of the third proposed algorithm for $\alpha = 0.1$.

Dataset	C_1	C_2	*Purity*	*Mirkin*	*F-measure*	*VI*	*PC*	*V-measure*
Diabetic	935	216	0.5673	0.4909	0.5799	0.1644	0.2566	1.0002
Phishing	705	10350	0.5952	0.4819	0.7187	0.0953	0.2983	1.0000
NSL-KDD	10527	137990	0.5457	0.4958	0.6270	0.0736	0.3538	1.0000
Banknote authentication	690	682	0.9475	0.0994	0.9474	0.0472	0.4528	1.0001
Spam	1733	2868	0.7022	0.4182	0.6679	0.1416	0.2887	1.0001
Covertype	2319	578693	**0.9647**	**0.0753**	**0.9764**	**0.0135**	**0.4757**	1.0000

Table 7. Performance evaluation of the third proposed algorithm for $\alpha = 0.2$.

Dataset	C_1	C_2	*Purity*	*Mirkin*	*F-measure*	*VI*	*PC*	*V-measure*
Diabetic	876	275	0.5804	0.4871	0.5634	0.1728	0.2580	1.0003
Phishing	786	10269	0.6025	0.4790	0.7203	0.0963	0.3051	1.0000
NSL-KDD	11197	137320	0.5495	0.4951	0.6238	0.0742	0.3529	1.0000
Banknote authentication	692	680	0.9490	0.0968	0.9488	0.0463	0.4538	1.0001
Spam	1683	2918	0.7005	0.4196	0.6617	0.1415	0.2875	1.0001
Covertype	6482	574530	**0.9647**	**0.0880**	**0.9663**	**0.0161**	**0.4758**	1.0000

Table 8. Performance evaluation of the third proposed algorithm for $\alpha = 0.3$.

Dataset	C_1	C_2	*Purity*	*Mirkin*	*F-measure*	*VI*	*PC*	*V-measure*
Diabetic	836	315	0.5786	0.4876	0.5567	0.1783	0.2567	1.0003
Phishing	753	10302	0.5995	0.4802	0.7196	0.0959	0.3024	1.0000
NSL-KDD	11726	136791	0.5525	0.4945	0.6214	0.0747	0.3521	1.0000
Banknote authentication	698	674	0.9534	0.0889	0.9532	0.0435	0.4570	1.0001
Spam	1584	3017	0.7033	0.4173	0.6556	0.1399	0.2878	1.0001
Covertype	6898	574114	**0.9647**	**0.0893**	**0.9652**	**0.0163**	**0.4756**	1.0000

Table 9. Performance evaluation of the third proposed algorithm for $\alpha = 0.4$.

Dataset	C_1	C_2	*Purity*	*Mirkin*	*F-measure*	*VI*	*PC*	*V-measure*
Diabetic	802	349	0.5769	0.4882	0.5507	0.1823	0.2559	1.0003
Phishing	830	10225	0.6065	0.4773	0.7211	0.0968	0.3084	1.0000
NSL-KDD	12292	136225	0.5556	0.4938	0.6188	0.0753	0.3512	1.0000
Banknote authentication	704	668	0.9577	**0.0810**	0.9576	0.0406	0.4604	1.0001
Spam	1535	3066	0.7018	0.4186	0.6487	0.1395	0.2870	1.0001
Covertype	7473	573539	**0.9647**	0.0910	**0.9638**	**0.0167**	**0.4758**	1.0000

Table 10. Performance evaluation of the third proposed algorithm for $\alpha = 0.5$.

Dataset	C_1	C_2	*Purity*	*Mirkin*	*F-measure*	*VI*	*PC*	*V-measure*
Diabetic	787	364	0.5778	0.4879	0.5474	0.1837	0.2559	1.0004
Phishing	853	10202	0.6086	0.4764	0.7215	0.0971	0.3100	1.0000
NSL-KDD	13089	135428	0.5601	0.4928	0.6152	0.0759	0.3505	1.0000
Banknote authentication	706	666	0.9592	**0.0783**	0.9591	0.0396	0.4615	1.0001
Spam	1495	3106	0.7009	0.4193	0.6433	0.1391	0.2865	1.0001
Covertype	8212	572800	**0.9647**	0.0933	**0.9621**	**0.0171**	**0.4759**	1.0000

Table 11. Performance evaluation of the third proposed algorithm for $\alpha = 0.6$.

Dataset	C_1	C_2	*Purity*	*Mirkin*	*F-measure*	*VI*	*PC*	*V-measure*
Diabetic	769	382	0.5812	0.4868	0.5426	0.1850	0.2564	1.0004
Phishing	1011	10044	0.6071	0.4770	0.7154	0.1017	0.2917	1.0000
NSL-KDD	14228	134289	0.5667	0.4911	0.6100	0.0768	0.3499	1.0000
Banknote authentication	709	663	0.9614	**0.0743**	**0.9613**	0.0380	0.4632	1.0001
Spam	1466	3135	0.7025	0.4180	0.6419	0.1384	0.2871	1.0001
Covertype	9261	571751	**0.9647**	0.0964	0.9595	**0.0176**	**0.4758**	1.0000

Table 12. Performance evaluation of the third proposed algorithm for $\alpha = 0.7$.

Dataset	C_1	C_2	*Purity*	*Mirkin*	*F-measure*	*VI*	*PC*	*V-measure*
Diabetic	754	397	0.5838	0.4859	0.5441	0.1860	0.2568	1.0004
Phishing	1214	9841	0.6034	0.4786	0.7064	0.1070	0.2765	1.0000
NSL-KDD	15980	132537	0.5764	0.4883	0.6023	0.0781	0.3485	1.0000
Banknote authentication	712	660	0.9621	**0.0729**	**0.9620**	0.0385	0.4636	1.0001
Spam	1441	3160	0.7018	0.4186	0.6382	0.1381	0.2868	1.0001
Covertype	10733	570279	**0.9647**	0.1008	0.9559	**0.0184**	**0.4759**	1.0000

Table 13. Performance evaluation of the third proposed algorithm for $\alpha = 0.8$.

Dataset	C_1	C_2	*Purity*	*Mirkin*	*F-measure*	*VI*	*PC*	*V-measure*
Diabetic	747	404	0.5864	0.4851	0.5497	0.1863	0.2572	1.0004
Phishing	1428	9627	0.6013	0.4795	0.6977	0.1117	0.2688	1.0000
NSL-KDD	18888	129629	0.5927	0.4828	0.5900	0.0799	0.3469	1.0000
Banknote authentication	721	651	0.9643	**0.0689**	**0.9642**	0.0387	0.4651	1.0001
Spam	1426	3175	0.7016	0.4187	0.6361	0.1379	0.2867	1.0001
Covertype	13308	567704	**0.9647**	0.1085	0.9497	**0.0197**	**0.4760**	1.0000

Table 14. Performance evaluation of the third proposed algorithm for $\alpha = 0.9$.

Dataset	C_1	C_2	*Purity*	*Mirkin*	*F-measure*	*VI*	*PC*	*V-measure*
Diabetic	741	410	0.5882	0.4844	0.5539	0.1865	0.2575	1.0004
Phishing	1925	9130	0.5959	0.4816	0.6769	0.1208	0.2605	1.0000
NSL-KDD	24685	123832	0.6249	0.4688	0.5674	0.0825	0.3455	1.0000
Banknote authentication	741	631	**0.9672**	**0.0634**	**0.9672**	0.0389	0.4673	1.0001
Spam	1397	3204	0.7027	0.4178	0.6339	0.1371	0.2872	1.0001
Covertype	20218	560794	0.9647	0.1288	0.9330	**0.0228**	**0.4762**	1.0000

Table 15. Performance evaluation of the third proposed algorithm for $\alpha = 1$.

Dataset	C_1	C_2	*Purity*	*Mirkin*	*F-measure*	*VI*	*PC*	*V-measure*
Diabetic	729	422	0.5899	0.4838	0.5601	0.1872	0.2578	1.0004
Phishing	8319	2736	0.5597	0.4929	0.5645	0.1333	0.2530	1.0000
NSL-KDD	130195	18322	0.5979	0.4808	0.5784	0.0837	0.3036	1.0000
Banknote authentication	765	607	0.9541	**0.0876**	**0.9541**	0.0515	0.4559	1.0001
Spam	1380	3221	0.7020	0.4184	0.6310	0.1369	0.2870	1.0001
Covertype	574292	6720	**0.9647**	0.0880	0.0787	**0.0163**	**0.4671**	1.0000

In addition, at $\alpha = 0.6$, $\alpha = 0.7$, $\alpha = 0.8$, $\alpha = 0.9$ and $\alpha = 1$, according to the Mirkin and F-measure metrics, the best indicators for the Banknote authentication dataset were achieved.

V-measure does not have a discriminating ability, i.e. its value on different datasets is almost the same for the methods. From this, it can be concluded that the use of the V-measure is not useful for evaluating the results of clustering. Therefore, in the following comparisons, it was not considered.

To illustrate the viability of the developed anomaly detection algorithms, the results are presented in Figure 1-5. Figure 1 shows the results of the third implemented clustering algorithm for the purity metric.

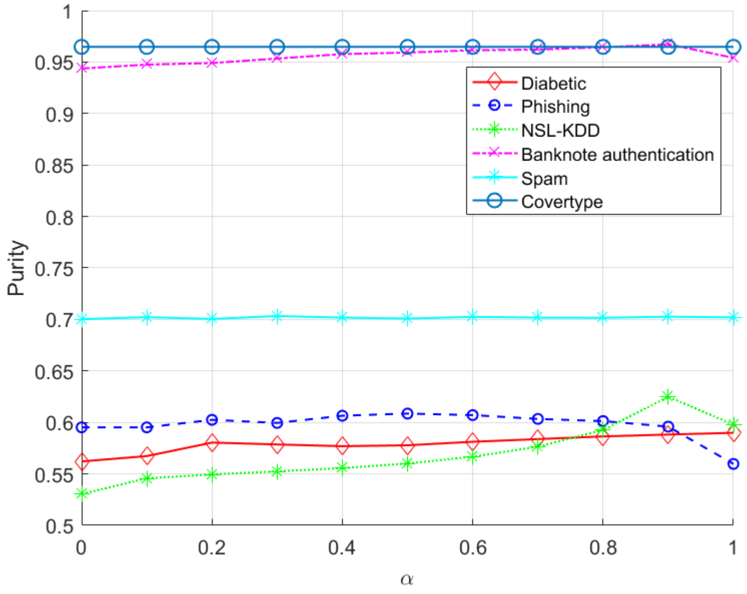


Figure 1. The influence of α values on the purity metric for different datasets.

Based on the experimental results, it can be concluded that the proposed approach shows the best results for Covertype and Banknote authentication datasets. The lowest values were observed for NSL-KDD, but at $\alpha = 0.9$, an improvement can be seen.

The lowest results for the Mirkin metric for all α values were obtained for Covertype and Banknote authentication datasets (Figure 2). For the Spam dataset, the value of the metric practically did not change and was $\sim$41.8%.

In Figure 3, it can be seen that the highest results were obtained for the Covertype dataset at α from 0 to 0.9, while at $\alpha = 1$ the value dropped sharply and amounted to 7.87%. The worst results according to the F-measure metric were obtained for the Diabetic dataset.

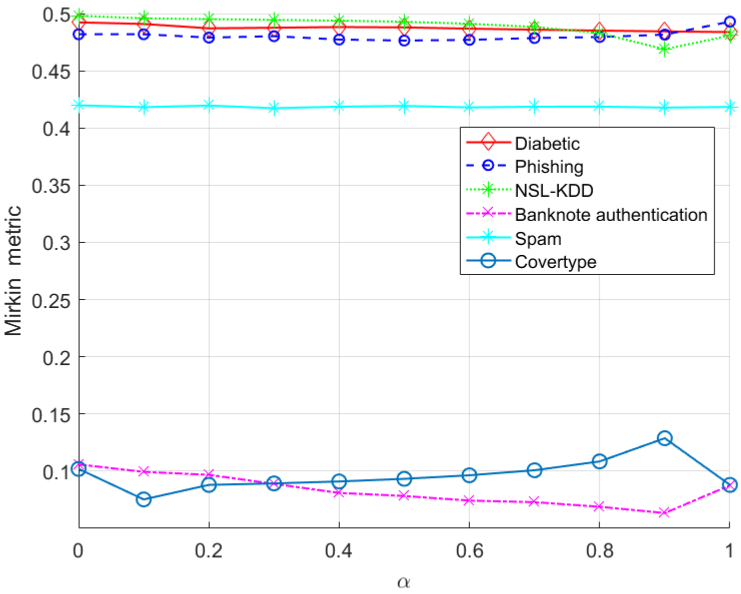


Figure 2. The influence of α values on the Mirkin metric for different datasets.

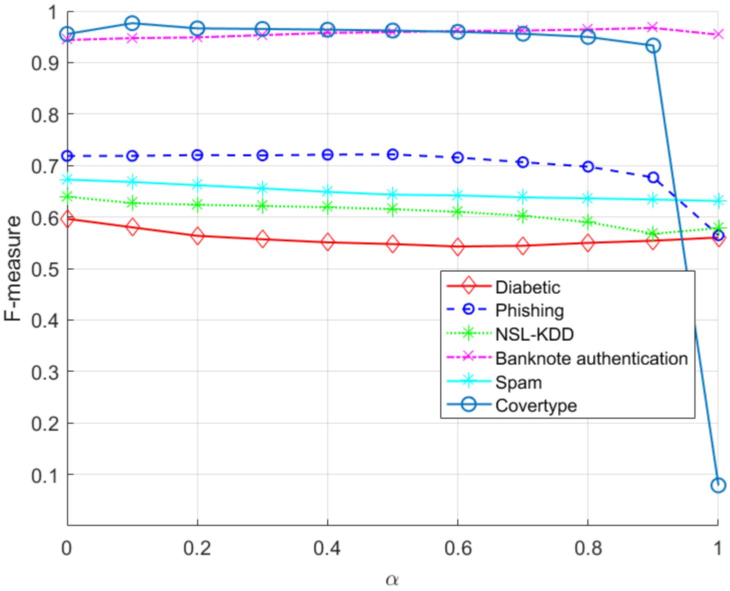


Figure 3. The influence of α values on the F-measure metric for different datasets.

The best results, i.e. the minimum values of the VI metric were obtained for the Covertype dataset, Banknote authentication dataset and NSL-KDD (Figure 4). Values of VI worsened with the increase of α value for the Diabetic dataset.

From Figure 5, according to the experimental results, it was obtained that Covertype and Banknote authentication datasets achieve the best results for almost all values of α. For Spam dataset, the metric value was practically

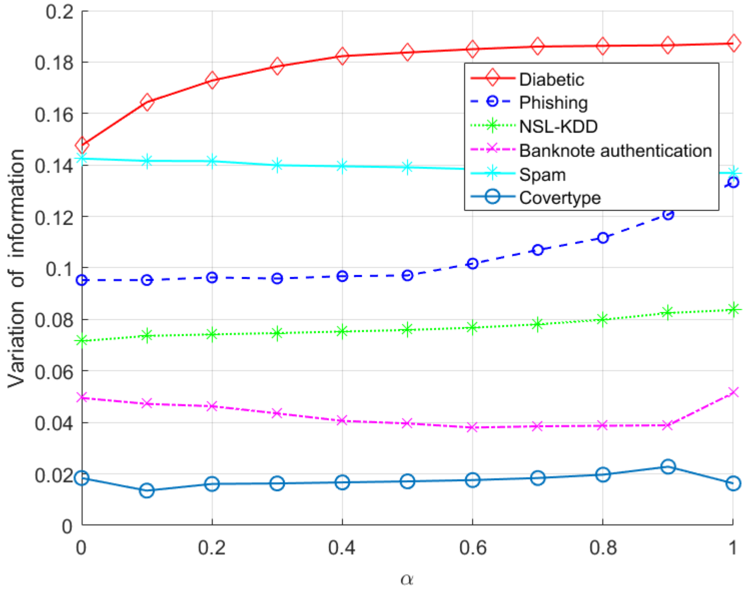


Figure 4. The influence of α values on the variation of information for different datasets.

constant. The values of the PC metric fall sharply at α from 0.6 to 1 for Phishing dataset, and at α=1 for NSL-KDD.

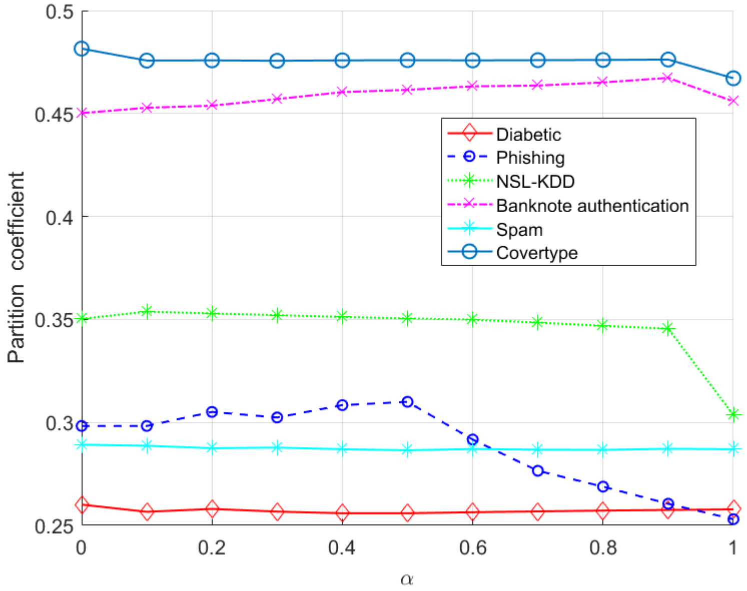
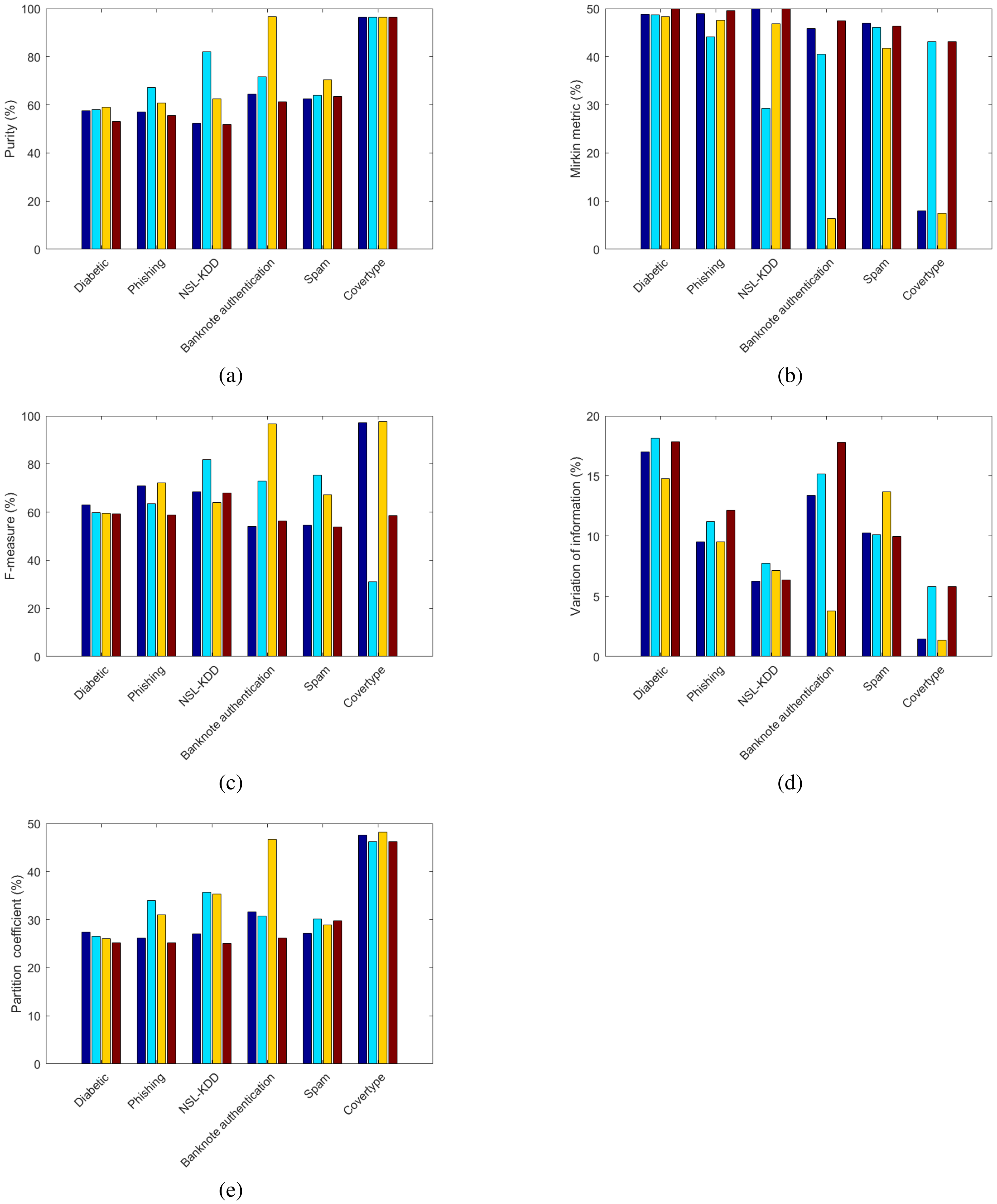


Figure 5. The influence of α values on the partition coefficient for different datasets.

A comparison of the evaluation metrics' values for the first (dark blue bars), second (blue bars), third (yellow bars) proposed algorithms and the k-means algorithm (red bars) on six datasets is more clearly illustrated in Figure 6. For the third algorithm, the best results for each metric at every α were selected.



(a)

(b)

(c)

(d)

(e)

Figure 6. The comparison of the proposed algorithms with the k-means algorithm based on evaluation metrics.

Based on the experimental results, it can be concluded that the first and third proposed approaches are superior to the k-means algorithm in all evaluation metrics for the Covertype dataset. Purity, Mirkin, F-measure and PC metrics showed good results for NSL-KDD dataset when applying the second algorithm. According to all metrics, Banknote authentication dataset achieved the best result for the third algorithm, Phishing dataset – for the second algorithm, and Diabetic dataset – for the first and third algorithms.

The values of Purity, Mirkin, and F-measure have shown the best results for the second and third algorithms, while PC metric was the best only for the second algorithm on Spam dataset. It can be concluded that the third algorithm works well on datasets of small and large sizes, while the second algorithm has shown the best results on the datasets of medium size.

6. Conclusion

In this paper, new clustering algorithms were proposed for anomaly detection in Big data. The aim of the algorithms presented in the paper is to improve the anomaly detection. The algorithms that minimize the compactness of clusters and maximize the separation of clusters from each other according to the distances between their centers and the remoteness of cluster centers from the selected common center of points in the dataset were presented. The comparison was made using six datasets containing anomalous values. The quality of the clustering result was estimated using six evaluation metrics. An important feature of the proposed approaches is that they increase the accuracy of anomalous values detection based on clustering. The performance of the proposed algorithms with the k-means algorithm was compared. It can be concluded that the proposed algorithms work efficiently on real datasets of different size.

It is important that the proposed approaches can be applied in various research fields. Future research will focus on the development and application of ensembles of clustering algorithms to anomaly detection.

Acknowledgement

This work was supported by the Science Development Foundation under the President of the Republic of Azerbaijan – Grant EİF-KETPL-2-2015-1(25)-56/05/1.

Conflict of Interests

The authors declare that there is no conflict of interest with respect to research, authorship and publication of this paper.

REFERENCES

1. R. M. Alguliyev, R. M. Aliguliyev, A. Bagirov, and R. Karimov, *Batch clustering algorithm for Big data sets*, in Proc. AICT Conference, Baku, 2016.
2. F. Jiang, G. Liu, J. Du, and Y. Sui, *Initialization of K-modes clustering using outlier detection techniques*, Information Sciences, vol. 332, pp. 167–183, 2016.
3. G. S. D. S. Jayakumar and B. J. Thomas, *A new procedure of clustering based on multivariate outlier detection*, Journal of Data Science, vol. 11, no. 1, pp. 69–84, 2013.
4. F. Macia-Perez, J. Berna-Martinez, A. Fernandez, and M. Abreu, *Algorithm for the detection of outliers based on the theory of rough sets*, Decision Support Systems, vol. 75, pp. 63–75, 2015.
5. T. S. Xu, H. D. Chiang, G. Y. Liu, and C. W. Tan, *Hierarchical k-means method for clustering large-scale advanced metering infrastructure data*, IEEE Transactions on Power Delivery, vol. 32, no. 2, pp. 609–616, 2017.
6. B. Liu, Y. Xiao, P. S. Yu, Z. Hao, and L. Cao, *An efficient approach for outlier detection with imperfect data labels*, IEEE Trans. Knowl. Data Eng., vol. 26, no. 7, pp. 1602–1616, 2014.
7. A. M. C. Souza and J. R. A. Amazonas, *An outlier detect algorithm using big data processing and internet of things architecture*, Procedia Computer Science, vol. 52, pp. 1010–1015, 2015.

8. J. Huang, Q. Zhu, and L. Y. J. Feng, *A non-parameter outlier detection algorithm based on Natural Neighbor*, Knowl.-Based Syst., vol. 92, pp. 71–77, 2016.
9. M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander, *LOF: identifying density-based local outliers*, ACM Sigmod Record, vol. 29, no. 2, pp. 93–104, 2000.
10. J. Ha, S. Seok, and J.-S. Lee, *Robust outlier detection using the instability factor*, Knowl.-Based Syst., vol. 63, pp. 15–23, 2014.
11. M. Capó, A. Pérez, and J. A. Lozano, *An efficient approximation to the k-means clustering for massive data*, Knowl.-Based Syst., vol. 117, pp. 56–69, 2017.
12. L. Ott, L. Pang, F. Ramos, D. Howe, and S. Chawla, *Integer programming relaxations for integrated clustering and outlier detection*, In arXiv:1403.1329, 2014.
13. G. Gan and M. Ng, *k-means clustering with outlier removal*, Pattern Recognition Letters, vol. 90, pp. 8–14, 2017.
14. R. Dave and R. Krishnapuram, *Robust clustering methods: a unified view*, IEEE Trans. Fuzzy Syst., vol. 5, no. 2, pp. 270–293, 1997.
15. J. Eggermont, J. N. Kok, and W. A. Kosters, *Genetic programming for data classification: partitioning the search space*, ACM SAC Symposium, pp. 1001–1005, 2004.
16. M. Lichman, *UCI Machine Learning Repository*, University of California, Available at http://archive.ics.uci.edu/ml, 2013.
17. P. Aggarwal and S. K. Sharma, *Analysis of KDD dataset attributes-class wise for intrusion detection*, Proc. Comp. Sci., vol. 57, pp. 842–851, 2015.
18. B. Antal and A. Hajdu, *An ensemble-based system for automatic screening of diabetic retinopathy*, Knowl.-Based Syst., vol. 60, pp. 20–27, 2014.
19. E. Decenciere, X. Zhang, G. Cazuguel, B. Lay, B. Cochener, C. Trone, P. Gain, R. Ordonez, P. Massin, A. Erginay, B. Charton, and J.-C. Klein, *Feedback on a publicly distributed database: the messidor database*, Image Analysis & Stereology, vol. 33, pp. 231–234, 2014.
20. J. McHugh, *Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln laboratory*, ACM Trans. Inf. and Syst. Sec., vol. 3, pp. 262–294, 2000.
21. J. A. Blackard and J. D. Denis, *Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables*, Comp. and Elect. Agriculture, vol. 24, pp. 131–151, 2000.
22. R. Mohammad, F. A. Thabtah, and T. L. McCluskey, *Phishing websites dataset*, University of Huddersfield, Available at https://archive.ics.uci.edu/ml/datasets/Phishing+Websites, 2015.
23. R. M. Alguliev, R. M. Aliguliyev, T. Kh. Fataliyev, and R. Sh. Hasanova, *Weighted consensus index for assessment of the scientific performance of researchers*, COLLNET J. Scientometrics and Inf. Management, vol. 8, pp. 371–400, 2014.
24. F. Boutin and M. Hascoet, *Cluster validity indices for graph partitioning*, in Proc. ICIV Conference, pp. 376–381, 2004.
25. A. M. Rubinov, N. V. Soukhorukova, and J. Ugon, *Classes and clusters in data analysis*, Euro. J. Operational Research, vol. 173, pp. 849–865, 2006.
26. B. Mirkin, *Mathematical classification and clustering*, J. Global Optimization, vol. 12, pp. 105–108, 1998.
27. J. C. Bezdek and N. R. Pal, *Some new indexes of cluster validity*, IEEE Trans. Syst., Man and Cyber, Part B, vol. 28, pp. 301–315, 1998.
28. A. Patrikainen and M. Meila, *Comparing subspace clusterings*, IEEE Trans. Knowl. and Data Engin., vol. 18, pp. 902–916, 2006.
29. A. Rosenberg and J. Hirschberg, *V-measure: a conditional entropy-based external cluster evaluation measure*, in Proc. EMNLP-CoNLL Conference, pp. 410–420, 2007.
30. R. M. Aliguliyev, *Performance evaluation of density-based clustering methods*, Inf. Sci., vol. 179, pp. 3583–3602, 2009.
31. I. Eyal, I. Keidar, and R. Rom, *Distributed data clustering in sensor networks*, Distrib. Comput., vol. 24, pp. 207–222, 2010.