JOURNAL OF APPLIED
COMPUTER SCIENCE
Vol. 25 No. 2 (2017), pp. 53-68

Outlier Mining Using the DBSCAN Algorithm

Agnieszka Nowak-Brzezińska[1], Tomasz Xięski[1]

[1]*Institute of Computer Science*
University of Silesia in Katowice
Bankowa 12, 40-007 Katowice, Poland
agnieszka.nowak@us.edu.pl

Abstract. *This paper introduces an approach to outlier mining in the context of a real-world dataset containing information about the mobile transceivers operation. The goal of the paper is to analyze the influence of using different similarity measures and multiple values of input parameters for the density-based clustering algorithm on the number of outliers discovered during the mining process. The results of the experiments are presented in section 4 in order to discuss the significance of the analyzed parameters.*
Keywords: *outlier detection, similarity analysis, clustering, knowledge-based systems..*

1. Introduction

Outlier detection is a fundamental issue in data mining, it has been specifically used to detect and remove anomalous objects from data. Data mining, in general, deals with the discovery of nontrivial, hidden and interesting knowledge from different types of data. With the development of information technologies, the number of databases and their dimensions and complexity, has grown rapidly. One of the basic problems of data mining is outlier detection. The identification of an outlier is affected by various factors, many of which have become the subject of practical applications such as in the public health or finance field. In the first case

(public health), outlier detection techniques help to detect anomalous patterns in patient medical data which could be symptoms of an ailment. In the second example, outlier detection methods can identify suspicious credit card transactions. Generally, outliers are the points which are different from or inconsistent with the rest of the data. It can be novel, new, abnormal, unusual or noisy information thus it is often more interesting than the majority of the actual data.

The article presents the analysis of the influence of different clustering parameters on the results of the final clusters' structure and their ability to mine outliers. Another important issue addressed in the experiments is the sampling of the dataset and its effect on the clustering structure.

1.1. The structure of the article

The paper is organized as follows. In the next section related approaches to outlier detection and their classification is presented. Moreover, the motivation for detecting outliers in a real-world dataset containing information about the mobile transceivers operation was stated.

In section 3 the reasons for selecting the DBSCAN density-based algorithm were introduced. What is more the aspects of choosing optimal initial parameters and the notion of clustering quality was discussed.

The next section focuses on the carried out experiments. The structure of the dataset was described as well as the methodology of all experiments was presented. All experimental results were commented in detail with regards to the clustering structure, its quality and outlier presence.

The last section presents the summary and conclusions from the performed experiments.

2. Related works

In the literature, there are a number of extensive reviews discussing anomaly detection approaches [1, 2, 3]. A recent review of the anomaly detection problems, techniques, and application areas is presented in [4, 5]. The anomaly detection techniques can be classified as statistical approaches and distance-based approaches. The aim of the first type of techniques is to develop a statistical model of the data and identify data that does not fit into the model, whereas the aim of the second type (distance-based approaches) is to measure the distance between data – anomalies are data for which the distance is greater than some

given threshold. An example of an algorithm from the distance-based approaches is *DBSCAN* (Density-Based Spatial Clustering of Applications with Noise) [6] -– one of the most popular and effective algorithm for finding anomalies. The aim of the *DBSCAN* algorithm is to discover abnormal points that do not fit any of the clusters.

The authors for last few years have been working on real-life data containing information about the mobile transceivers operation. The functioning of those devices is regulated by a particular controller mounted typically in a base station. In [7] the authors included the results of research which was based on the detection of the most problematic transceivers (characterized by a high average level of unavailability and high number of registered events) using clustering algorithms and visualization techniques. Based on the results, the mobile telephony provider can optimize the network structure, which should directly translate into improving the quality of offered services. Thus the possibility of detecting outliers as soon as possible is so important in this area.

3. The DBSCAN algorithm

Authors of this paper have selected the *DBSCAN* density-based algorithm as a basis for discovering trends and relations between objects (like network devices). This method has several advantages over traditional hierarchical or partitioning approaches like: the possibility to discover groups of irregular shapes and sizes, resistance to outliers and a relatively low computational complexity[1]. Also preliminary experiments on a dataset gathering information about mobile transceivers (described in detail in [9]) confirmed that it is possible to apply the mentioned technique with success in an information retrieval task[2]. Unfortunately, when dealing with large volumes of data, the *DBSCAN* algorithm can also create a large number of clusters, which makes their analysis difficult (in the context of knowledge discovery or extraction). That is why the research process should be supported by the usage of clustering visualization methods, which were analyzed by the authors in pararell. The results of the research on using different techniques for clusters visualization were included in previous authors' research [7].

[1]When using index structures, like R-trees[8], the average computational complexity is about $O(\log n)$, where n is the number of instances [6].

[2]Other approaches to the problem of clustering large volumes of complex data were discussed by authors of this paper in [10].

3.1. Parameter estimation

Every data mining task has the problem with choosing the right initial values of input parameters. Every parameter influences the algorithm in a specific way. For the *DBSCAN* algorithm, two parameters – ϵ and *MinPts* – are needed. The first one (ϵ) has a big impact on how coherent the clusters are (that is how similar are the objects inside one cluster) and the second parameter (*MinPts*) controls how big the created clusters should be. The main idea of the *DBSCAN* algorithm is based on the ϵ-neighborhood concept (see definition 1).

Definition 1 *The ϵ-neighborhood of object p (denoted by $N_\epsilon(p)$) is defined as:*

$$N_\epsilon(p) = q \in D | dist(p,q) \leq \epsilon, \tag{1}$$

where D is the dataset, $dist(p,q)$ states the dis-similarity between object p and q, whereas ϵ is the maximum neighborhood radius. The ϵ-neighborhood of object p thus are all those objects q, which distance from p is less or equal than the given threshold ϵ.

The definition 1 can lead to a conclusion that to form a valid cluster by using the *DBSCAN* algorithm, there should exist at least *MinPts* objects in a given ϵ-neighborhood. This is only partially true, as one can distinguish two types of objects in a cluster: a so called *core* and *border point*. Core points are objects in the center of the cluster and border points are objects on its border. It is obvious that the ϵ-neighborhood of a core point contains more objects than the ϵ-neighborhood of a border point. And therefore the *DBSCAN* algorithm uses the notions of *density-reachable* and *density-connected* points to form a cluster. This definitions are discussed in detail in [6], but still relay greatly on the values of ϵ and *MinPts* which must be specified by the user.

Another very important aspect is the distance or similarity measure between objects used in every step of the clustering algorithm. The choice of a particular distance function is tightly coupled to the choice of ϵ values, and has a major impact on the results. In general, it will be necessary to first identify a reasonable measure of dissimilarity for the dataset, before the optimal values of ϵ parameter can be chosen. In this research two, well known distance measures are used: *Gower* and *Hamming*. The first mentioned measure can be expressed as follows:

$$GowerDistance(p_i, p_j) = 1 - \frac{\sum_{k=1}^{n} s_{ijk} w_{ijk}}{\sum_{k=1}^{n} w_{ijk}}, \tag{2}$$

where p_j and p_i are analyzed objects, n is the number of attributes, w_{ijk} expresses the weight connected with the particular attribute and s_{ijk} is dependent on the attribute type. The weight (w_{ijk}) is always set to 1, because all the attribute values are known and have the same importance in the clustering process. For qualitative the s_{ijk} equals 1 if compared objects do not differ by the analyzed attribute value or 0 otherwise. For quantitative data s_{ijk} is expressed as:

$$s_{ijk} = 1 - \frac{|x_{ik} - x_{jk}|}{R_k}, \tag{3}$$

where x_{ik} and x_{jk} are values of the k-th attribute for both objects, and R_k is the difference between the maximal and minimal value of the k-th attribute.

The Hamming distance was defined as follows:

$$HammingDistance(p_i, p_j) = \sum_{k=1}^{n} s_{ijk} \tag{4}$$

where p_j and p_i are analyzed objects, n is the number of attributes, and s_{ijk} equals 0 if compared objects do not differ by the analyzed attribute value or 1 otherwise.

Therefore, the main goal of this research is based on the evaluation of different values of the $DBSCAN$ input parameters which have an influence on the clustering efficiency. The aspects analyzed in this research are as follows:

- clustering parameters for $DBSCAN$ algorithm: ϵ and $MinPts$,
- different similarity measures.

3.2. The pseudocode of DBSCAN algorithm

The general operating principle of the $DBSCAN$ algorithm can be presented in following points:

1. Select an object p from the dataset.
2. Determine all density-reachable objects from the current one, with regard to the ϵ and $MinPts$ parameters:
   (a) if p is a core point, form a cluster,

(b) if p is a border point and no other object is density-reachable from p, select another object from the dataset (and continue from the second step).

3. Continue from the first step until all objects from the dataset are analyzed.

The first step in the algorithm is to draw the object p, and to designate all objects that are density-reachable from the object p (given ϵ – the maximum radius of the neighborhood and $MinPts$ – the minimum number of objects in a group). If p can be considered as a core point, this results in the creation of the first cluster. If p is a border point, that means that no object is density-reachable from p and so the algorithm chooses another object from the dataset. This process is repeated until all objects from the input data collection are analyzed. Objects not classified to any cluster are marked as information noise [6].

3.3. The quality of created clusters

In order to determine the optimal input parameters for the density-based algorithm, several clusterings were created, each with different values of ϵ and $MinPts$ parameters. The quality of the generated clusterings was rated based on the following cluster evaluation measure:

$$clustering\ quality = \frac{\sum_{i=1}^{m} \frac{\sum_{p \in C_i} dist(p,u_i)}{|A| \cdot |C_i|}}{m} \tag{5}$$

where m – number of generated clusters, C_i – ith cluster, $dist(p, u_i)$ – distance [11] between an object p (belonging to cluster C_i) and the cluster's representative u_i, $|A|$ – number of attributes in the dataset, $|C_i|$ – number of objects belonging to cluster C_i.

The formula expressed in equation 5 measures cluster cohesion. Values closer to zero represent a better clustering (in terms of overall quality), whereas values closer to one designate the opposite. Because the density based algorithm can (for specific values of input parameters) generate clusters consisting of single objects (which can be regarded as outliers), in such case the distance of the object to its cluster representative is set as maximum. This way, the formula will not promote such clusters (consisting of only one object).

Cluster cohesion is only one of several other internal[3] cluster validity mea-

[3]Internal indexes (criteria) are used to measure the quality of a clustering structure without respect to external information like externally supplied class labels.

sures. One could also measure separation (to detect how distinct or well-separated a cluster is from others), but authors believe that cohesion is even more important, because it allows to check whether the data (objects) within groups are really well assigned to clusters – if there is a sufficient level of similarity between objects belonging to the same cluster. In the domain literature [11, 4] there are defined several cluster validity measures based on cohesion and separation (like Sum of Squared Error or the Silhouette Coefficient) and could be used in addition to the measure presented in equation 5.

4. Experiments

In this section the structure of the analyzed cell_loss dataset and the methodology of the carried out experiments were described. Furthermore the obtained experiment results were presented and discussed in detail.

4.1. Structure of the cell_loss dataset

The dataset being analyzed in this work for outliers contains information about the mobile transceivers operation. Those devices are called cells and their functioning is regulated by a particular controller mounted typically in a base station. The information was gathered for a period of 9 months, starting April 2010. This resulted in creating 143486 objects (records), described by 19 attributes, saved in one database table. As such, each cell has assigned a particular vendor, controller, geographic location and its degree of availability, work orders (like for maintenance) and measurement date and time tracked. The structure of one data record was described in detail in our previous work [7]. For the purpose of the experiments carried out in this work, 14 attributes were selected in the clustering process (based on the domain experts suggestion) – the cell and event identifiers, start and end event times and degree of inaccessibility expressed as a real number were omitted. More details for an analyzed dataset is included in the [7].

Another important issue addressed in the experiments is the sampling of the dataset. Several clustering algorithms (including *CLARA* [11] and *CURE* [12]) reduce the size of input by drawing a random sample from the entire dataset to discover groups in large datasets. Unfortunately the reduction in input data due to sampling can affect the efficiency of a cluster analysis algorithm (in terms of clustering quality). Even the creators of *CURE* state that „since we do not consider the entire dataset, information about certain clusters may be missing from the input.

As a result, our clustering algorithms may miss out certain clusters or incorrectly identify certain clusters" [12].

4.2. The methodology

The goal of the experiments was to check whether choosing of different similarity measures and/or clustering parameters influences the possibility of finding outlier-type data. Thus, in this section, experimental evaluation of two similarity measures and different values of two input parameters of the $DBSCAN$ algorithm (ϵ, $MinPts$) and their influence on the success of discovering outliers in data is presented. The experiments are provided for one dataset, which was preprocessed in order to get smaller pieces of data. In order to do that, the whole dataset (100% of data) is divided and compared to its 1%, 10%, 25% and 50% chunk. This step resulted in the creation of 5 different (considering their size) datasets. During the experiments, the authors examined the strength of the significant differences in the values of the number of created clusters, the number of discovered outliers, and others valueable information about the outliers in a given dataset, when a particular similarity measure and/or values of ϵ and $MinPts$ parameters were used.

To evaluate the effect of the ϵ and $MinPts$ parameters on discovering anomalies, we set them to different values and analyzed the results. As ϵ strongly depends on the similarity measure, its set of possible values is $\{1, 2, 3, 4\}$ for the *Hamming* measure or $\{0.1, \ldots, 0.9\}$ (with a step of 0.1) for the *Gower* similarity measure. The $MinPts$ value is always one of the following: $\{1, 2, 3\}$. It means that for each of the datasets (1%, 10%, 25%, 50% and 100%) there were 27 experiments for the *Gower*'s measure and 12 for the *Hamming*'s measure [4].

During the data mining process a lot of interesting information was recorded: how many outliers are in the data, what is their percent in the whole data, how many clusters are below the given threshold, how many objects are included in such clusters (below a given threshold), how many of them are so called singletons (a cluster with only one object inside of it) and how many small clusters contain at least two objects but less than a given threshold, and last but not least, the quality of created clusters. All these information were examined in this research and the results of the experiments are presented in Tables 1 and 2. The definition of the rows in Tables 1 and 2 are as follows:

[4] 27 is the number of all possible combinations calculated by using 9 values for ϵ and 3 values for $MinPts$, while 12 is the number of all possible variations of using 3 values for $MinPts$ and 4 values for ϵ.

- *Singleton* – number of clusters containing only one object,
- *Singleton%* the percentage of singleton clusters in the whole dataset,
- *#C_ltT* – the number of the clusters which size is below a given threshold,
- *#C_ltT%* – the percentage of the clusters which size is less than a given threshold,
- *#OiC_ltT* – the number of objects in clusters which size is less than a given threshold,
- *#OiC_ltT* – the percentage of the number of objects in clusters which size is less than a given threshold,
- *#C_ltTwoS* – the number of clusters which size is below a given threshold excluding singleton clusters,
- *#C_ltTwoS* – the percentage of the number of clusters which size is below a given threshold excluding singleton ones,
- *CQ* – the quality of created clusters.

4.3. Results

Table 1 presents the results of the comparison of using two different similarity measures for the *DBSCAN* algorithm used in the outlier mining process. *N* denotes the number of cases in which a given measure was used.

It is possible to see that there is a statistically significant difference between used similarity measures (*Gower*, *Hamming*) in values of such parameters like the number of created clusters (*#C*), the number of so-called singleton data (*Singleton*), the number of discovered anomalies (*Out*, *Out%*) as well as the number of clusters noted as too small (because their size is smaller than a given threshold T) to be a cluster (*#C_ltTwoS*) and the number of instances inside the clusters which are too small (*#OiC_ltT*). When the *Gower* measure is used, the smaller number of clusters as well as anomalies are created. This similarity measure leaves smaller number of instances as singletons after the clustering process (what means that more instances are noticed as similar to others and belong in one group).

Table 1: Similarity measures vs. examined clustering parameters for the cell_loss dataset

Parameter	Gower (N=108)	Hamming (N=87)
#C	71.74 ± 119.66	805.71 ± 1648.59
Out	57.12 ± 115.37	564.74 ± 1266.00
%Out	0.01 ± 0.04	0.04 ± 0.13
#C_ltTwoS	70.68 ± 119.68	801.16 ± 1642.65
#C_ltTwoS%	0.53 ± 0.47	0.79 ± 0.35
CQ	0.82 ± 0.23	0.75 ± 0.20
#OiC_ltT	283.69 ± 482.07	6393.54 ± 15180.32
#OiC_ltT%	0.02 ± 0.04	0.13 ± 0.21
T	577.25 ± 564.11	479.69 ± 433.31
#C_ltT	70.68 ± 1119.68	801.16 ± 1642.65
#C_ltT%	0.53 ± 0.47	0.79 ± 0.35
Singleton	20.04 ± 58.42	206.89 ± 655.34
Singleton%	0.004 ± 0.02	0.02 ± 0.08

What is also important, there is a statistically significant difference in the quality of created clusters between the results achieved for the *Gower* measure in comparison to the *Hamming* distance. It can be said that the usage of the *Gower* measure results in creating better clusters in the terms of quality (*CQ*). The comparison of using these two similarity measures is presented in the figure 1.

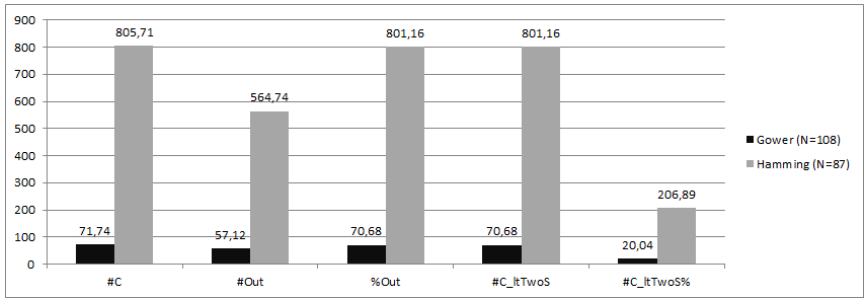

Figure 1: Comparison of results obtained by using the *Gower* and *Hamming* distance.

The aim of the second experiment (which results are presented in Table 2) was to determine the effect of sampling on the values of the size and the number of created clusters as well as the number of outliers discovered in the clustering process (which can be important e.g. in the task of searching through a cluster structure). In order to achieve this, four new datasets were created by reducing the original dataset (*cell_loss*) to respectively 1%, 10%, 25% and 50% of instances. To each one of the datasets a density based clustering algorithm was applied.

The only statistically significant difference can be seen in the quality of created clusters (CQ) and the number of instances in the clusters for which the size was smaller than a given threshold ($\#OiC\_ltT$). Obviously the number of instances in the dataset for which the clusters are built has a significant impact on the quality of created clusters: the smaller the dataset is the higher is its clustering quality. It can be easily explained. When we have a small number of instances from a given area, they are usually quite coherent, so the quality of the clusters which would be created from such data would be high enough. When the number of instances is too big, there is a bigger chance that some noise in data would appear, which definitely decreases the quality of the formed clusters. The explanation for achieving a smaller number of instances in the clusters smaller than a given threshold is also quite simple. The more instances we have in general to explore and to process, the more instances can be noticed as dissimilar to others and included in separate clusters what results in creating many small clusters (with size is smaller than a given threshold).

It was very important, during this research, to find the optimal values for initial clustering parameters as ϵ and $MinPts$. The authors, taking into account the specification of the domain data, decided to examine three values of the $MinPts$ parameter. Value higher than 3 would probably result in a greater number of singleton clusters, and thus the number of anomalies, discovered in such data. Having too many outliers discovered in data, it would be problematic to finally decide which of them are real outliers and which of them were marked as outliers because of poorly chosen input parameters (and not because they would really be anomalies).

Table 3 contains information about clusters and outliers, discovered by the DBSCAN algorithm and using different values of the $MinPts$ parameter. In the first row of the Table 3 values: 1,2 and 3 for parameter $MinPts$ are given.

The statistically significant difference between given values of the $MinPts$ parameter can be seen in the number of outliers ($\#Out$), the number of instances that created a singleton clusters ($Singleton$) as well as in the quality of the created clusters (CQ). The greater the value of the $MinPts$ is, the greater the number of

Table 2: The size of the dataset vs. examined clustering parameters

Parameter	Dataset size				
	1%	10%	25%	50%	100%
#C	34.78 ± 59.93	303.87 ± 812.84	426.79 ± 1122.97	452.31 ± 1214.49	649.79 ± 1684.24
#Out	56.04 ± 104.54	338.67 ± 909.02	383.64 ± 1118.35	330.04 ± 970.81	225.28 ± 781.24
%Out	0.04 ± 0.07	0.02 ± 0.06	0.01 ± 0.03	0.05 ± 0.16	0.001 ± 0.01
#OiC_ltT	70.41 ± 99.55	1058.64± 2508.52	2239.05± 5602.62	3129.31± 8884.75	7609.49± 19888.99
#OiC_ltT%	0.05 ± 0.07	0.07 ± 0.17	0.06 ± 0.16	0.09 ± 0.19	0.05 ± 0.14
#C_ltT%	33.52 ± 59.89	301.95 ± 810.91	424.10 ± 1118.85	448.96 ± 1209.51	646.56 ± 1677.93
#C_ltT	0.56 ± 0.45	0.64 ± 0.45	0.64 ± 0.45	0.71 ± 0.41	0.64 ± 0.45
T	14.0 ± 0.0	143.31 ± 0.47	359.31 ± 0.47	551.59 ± 301.17	1435.0 ± 0.0
#C_ltTwoS	33.52 ± 59.89	301.95 ± 810.91	424.10 ± 1118.85	448.96 ± 1209.51	646.56 ± 1677.93
#C_ltTwoS%	0.56 ± 0.45	0.64 ± 0.45	0.64 ± 0.45	0.71 ± 0.41	0.64 ± 0.45
CQ	0.93 ± 0.08	0.87 ± 0.12	0.81 ± 0.16	0.76 ± 0.21	0.62 ± 0.29
Singleton	19.56 ± 47.47	134.72 ± 519.36	139.92 ± 577.15	116.84 ± 461.41	76.03 ± 364.60
Singleton%	0.01 ± 0.03	0.01 ± 0.04	0.01 ± 0.02	0.02 ± 0.10	0.0005 ± 0.003

outliers (probably because it was impossible to find at least $MinPts$ neighbors for a given instance what results eventually in marking it as an outlier). The greater the number of $MinPts$ is, the higher the quality of created clusters (which is obvious).

To examine the second clustering parameter, which is ϵ, it is necessary to analyze the results of changing its values, separately for both analyzed similarity mea-

Table 3: The values of the *MinPts* parameter vs. examined clustering parameters

Parameter	MinPts value		
	1	2	3
#C	643.60 ± 1613.49	333.40 ± 948.10	220.62 ± 694.22
#Out	0.0 ± 0.0	310.2 ± 737.52	540.58 ± 1294.81
%Out	0.0 ± 0.0	0.03 ± 0.10	0.05 ± 0.13
#OiC_ltT	3291.40 ± 11102.86	2981.20 ± 10584.68	2756.26 ± 10133.85
#OiC_ltT%	0.09 ± 0.20	0.06 ± 0.14	0.05 ± 0.12
#C_ltT%	0.69 ± 0.44	0.65 ± 0.44	0.59 ± 0.43
#C_ltT	641.02 ± 1608.79	330.82 ± 943.43	217.92 ± 689.77
#C_ltTwoS	641.02 ± 1608.79	330.82 ± 943.43	217.92 ± 689.77
#C_ltTwoS%	0.69 ± 0.44	0.65 ± 0.44	0.59 ± 0.43
CQ	0.74 ± 0.23	0.74 ± 0.23	0.88 ± 0.15
Singleton	310.2 ± 737.52	0.0 ± 0.0	0.0 ± 0.0
Singleton%	0.03 ± 0.10	0.0 ± 0.0	0.0 ± 0.0

sures *Gower* and *Hamming*. The results for this research are included in Tables 4 and 5.

Both tables confirm the following conclusion: the greater the value of the ϵ parameter, the smaller the number of created clusters, discovered anomalies in data and all other parameters analyzed in the research. It can be also observed, that there is such a moment during changes of ϵ when the differences are strongly visible.

For the *Gower* measure, the most drastic change happens when the ϵ is increased from $0,3$ to $0,4$ and from $0,5$ to $0,6$. For the *Hamming* measure, such an important change is when we increase the ϵ from the value of 3 to 4.

The experiments described in this section were to confirm that the choice of optimal clustering parameter values is very important when we want to achieve a structure of coherent clusters with the usage of an density-based algorithm like the DBSCAN. The results presented in the Tables 1, 2, 3, 4 and 5 show that sometimes, badly chosen parameter values may produce too many small clusters which will be marked as outliers, even if they are not anomalies at all. On the other hand, choosing too large values of some clustering parameters, may produce too many

Table 4: The influence of ϵ on outlier mining efficiency – the *Gower* measure

ϵ	*#C*	*#Out*	*Singleton*	*#C_ltT*	*#OiC_ltT*	*#C_ltTwoS*	*CQ*
0.1	215.6 ± 135.48	187.80± 179.74	61.93 ± 101.45	214.13± 136.12	829.60 ± 450.38	214.13 ± 136.12	0.67± 0.19
0.2	213.53± 138.44	171.07± 163.88	61.93 ± 101.45	212.53± 138.44	822.73 ± 461.42	212.53 ± 138.44	0.67± 0.19
0.3	213.53± 138.44	171.07± 163.88	61.93 ± 101.45	212.53± 138.44	822.73 ± 461.42	212.53 ± 138.44	0.67± 0.19
0.4	12.73 ± 13.79	14.8 ± 18.71	5.67 ± 12.20	11.73 ± 13.79	22.80 ± 15.94	11.73 ± 13.79	0.72± 0.22
0.5	12.73 ± 13.79	14.80 ± 18.71	5.67 ± 12.20	11.73 ± 13.79	22.80 ± 15.94	11.73 ± 13.79	0.72± 0.22
0.6	1.20 ± 0.77	0.40 ± 1.06	0.20 ± 0.77	0.20 ± 0.77	0.20 ± 0.77	0.20±0.77	1.00± 0.0
0.7	1.07 ± 0.26	0.13 ± 0.35	0.07 ± 0.26	0.07 ± 0.26	0.07 ± 0.26	0.07±0.26	1.00± 0.0
0.8	1.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	1.0 ± 0.0
0.9	1.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	1.0 ± 0.0

Table 5: The influence of ϵ on outlier mining efficiency – the *Hamming* measure

ϵ	*#C*	*#Out*	*Singleton*	*#C_ltT*	*#OiC_ltT*	*#C_ltTwoS*	*CQ*
1	3430.40± 2579.54	2358.67± 2231.80	865.27 ± 1372.85	3414.60± 2572.18	27949.33± 26431.52	3414.60 ± 2572.18	0.66± 0.20
2	939.47 ± 659.48	631.20 ± 603.58	230.13 ± 370.09	932.67 ± 661.41	8195.00± 7834.77	932.67 ± 661.41	0.66± 0.19
3	138.27 ± 101.07	125.07 ± 128.70	46.60 ± 81.67	137.27 ± 101.07	445.93 ± 185.84	137.27 ± 101.07	0.67± 0.20
4	9.13 ± 12.90	11.73 ± 17.09	4.80 ± 11.35	8.13 ± 12.90	13.87 ± 14.23	8.13 ± 12.90	0.82± 0.16

clusters, which usually are small, and in the case of clustering large datasets it may hinder the task of searching through such a structure.

5. Summary

This article presents the evaluation of using a density-based DBSCAN algorithm in the context of outlier discovery in a real-world dataset containing information about the mobile transceivers operation. Related works in the field of outlier detection and the motivation for choosing the density-based algorithm were introduced. The issue of sampling of the dataset, optimal input parameter values selection and their impact on the resulting clustering structure was discussed.

The results from the experiments clearly show that in order to achieve a structure of coherent clusters, one has to carefully select the proper similarity measure (by considering multiple measures and examining the obtained results) as well as pay much attention in choosing the optimal values of input parameters for the given clustering algorithm. It is especially important because badly chosen parameter values may produce too many small clusters which will be marked as outliers, even if they are not anomalies at all.

References

[1] Aggarwal, C. C., *Outlier Analysis*, Springer, 2013.

[2] Aggarwal, C. C., *Data Mining - The Textbook*, Springer, 2015.

[3] Aggarwal, C. C. and Sathe, S., *Outlier Ensembles - An Introduction*, Springer, 2017.

[4] Tan, P.-N., Steinbach, M., and Kumar, V., *Introduction to Data Mining*, Addison-Wesley Longman Publishing Co., Inc., 2005.

[5] Zhang, J., *Advancements of Outlier Detection: A Survey*, EAI Endorsed Transactions on Scalable Information Systems, Vol. 13, No. 1, 2 2013.

[6] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X., *A density-based algorithm for discovering clusters in large spatial databases with noise*, In: In Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining, AAAI Press, 1996, pp. 226–231.

[7] Nowak-Brzezińska, A. and Xięski, T., *Exploratory Clustering and Visualization*, Procedia Computer Science, Vol. Volume/issue: 35C, 2014, pp. 1082–1091.

[8] Xia, T. and Zhang, D., *Improving the R*-tree with Outlier Handling Techniques*, In: Proceedings of the 13th Annual ACM International Workshop on Geographic Information Systems, GIS '05, ACM, New York, NY, USA, 2005, pp. 125–134.

[9] Wakulicz-Deja, A., Nowak-Brzezińska, A., and Xięski, T., *Efficiency of Complex Data Clustering*, In: Proceedings of the 6th International Conference on Rough Sets and Knowledge Technology, Springer-Verlag, 2011, pp. 636–641.

[10] Wakulicz-Deja, A., Nowak-Brzezińska, A., and Xięski, T., *Density-Based Method for Clustering and Visualization of Complex Data*, In: Proceedings of the 8th International Conference RSCTC 2012, Springer-Verlag, 2012, pp. 142–149.

[11] Han, J., Kamber, M., and Pei, J., *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers Inc., 2011.

[12] Guha, S., Rastogi, R., and Shim, K., *CURE: An Efficient Clustering Algorithm for Large Databases*, In: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, ACM, 1998, pp. 73–84.