Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 192 (2021) 3010–3019

Procedia Computer Science

www.elsevier.com/locate/procedia

25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Outliers in *Covid* 19 data based on Rule representation - the analysis of *LOF* algorithm.

Agnieszka Nowak - Brzezińska[a,*], Czesław Horyń[a]

[a] *University of Silesia in Katowice, Institute of Computer Science, Katowice, Bankowa 12, Poland*

Abstract

The article concerns the detection of outliers in rule-based knowledge bases containing data on *Covid* 19 cases. The authors move from the automatic generation of a rule-based knowledge base from source data by clustering rules in the knowledge base to optimize inference processes and to detecting unusual rules allowing for the optimal structure of rule groups. The paper presents a two-phase procedure, wherein in the first phase, we look for the optimal structure of rule clusters when there are outlier rules in the knowledge base. In the second phase, we detect outliers in the rules using the *LOF* (Local Outlier Factor) algorithm. Then we eliminate the unusual rules from the database and check whether the selected cluster quality measures are responded positively to the elimination of outliers, which would indicate that the rules were rightly considered outliers. The performed experiments confirmed the effectiveness of the *LOF* algorithm and selected cluster quality measures in the context of detecting atypical rules. The detection of such rules can support knowledge engineers or domain experts in knowledge mining to improve the completeness of the knowledge base, which is usually the basis of the decision support system.

Keywords: rules, knowledge base, outliers, LOF, quality indices, clustering ;

1. Introduction

Globally, as of 23 April 2021, there have been 144 358 956 confirmed cases of *Covid* 19 (in Americas there are 6513 486 confirmed Cases and in Europe 50 323 356), including 3 066 113 deaths, reported to WHO [19]. Every day there are new databases and new research works. Only one Science Direct database has 54 630 results when asked for *Covid* 19 WHO also collects in its database all the works that have been published so far on *Covid* 19. As of April 2021, it has over 10 000 of them. In this way, it collects information from sources such as PubMed, Medline, Elsevier, etc. [16]. Every few days Nature publishes new research papers on the fight against coronavirus (*Covid* 19 research updates) [17]. We all want to know the devastating virus in the best possible way and deal with it as soon as possible.

* Corresponding author. Tel.: +48-32-3689757 ; fax: +48-32-3689866.
E-mail address: agnieszka.nowak-brzezinska@us.edu.pl

1877-0509
10.1016/j.procs.2021.09.073

Our work aims to detect unusual hidden data (outliers) but in data as complex as decision rules. It is only a matter of time before intelligent expert applications will be widespread use, replacing a doctor who can quickly diagnose disturbing symptoms of the disease. These types of applications operate based on decision rules. These rules are either defined directly by domain experts or induced based on source data. Huge amounts of data are collected in databases every day.

Detecting outliers in rules in knowledge bases allows knowledge engineers to control better the consistency and completeness of the domain knowledge in decision support systems, where instead of an expert human being, a knowledge base and inference algorithms simulating the reasoning and knowledge of a human expert in the field are used in the decision-making process. We assume here that the rules are generated automatically, directly from the source data. There are many automatic rule generation algorithms; we will use decision rules generated according to the rough set theory, namely $LEM2$ algorithm [4]. When a machine performs this type of task instead of a human expert, the automatically generated rules may include unusual rules. Identification of such rules before implementing a given expert system will allow the knowledge engineer to ask human experts for the opportunity to supplement knowledge in a rare, so far insufficiently explored area.

One of the best-known algorithms for detecting outliers in data is the LOF (Local Outlier Factor) algorithm, which can detect even local outliers [6]. In our work, we want to check how well the LOF algorithm is doing in such specific data as rules in knowledge bases. These types of objects or structures are difficult to analyze for several reasons. First, they can be of different lengths because they can have a different number of premises. Secondly, they can contain various types of data: quantitative, qualitative, binary, etc. This also means that typical distance measures for immeasurable data cannot be used. We will then be supported by similarity measures that allow us to measure the closeness of qualitative data.

1.1. Related Works and Background

Anomaly detection algorithms are now used in many application domains and often enhance traditional rule-based detection systems. Applications such as intrusion detection, fraud detection, or unknown disease entities in medical applications are already quite well known. In recent years, we can observe more and more research on the application of outlier detection in data. In [13] the survey with a comprehensive overview of anomaly detection techniques related to the big data features of volume and velocity is provided. It has examined strategies for addressing the problem of high dimensionality. The authors of [12] propose two parallel local density-based algorithms, namely, $MRLOF$ (MapReduce based Local Outlier Factor) and $SLOF$ (Spark based Local Outlier Factor). Authors of [11] considered the detection of an outlier in qualitative data. In turn, the authors of the work of [10] focused on developing an algorithm for detecting singular outliers. Singular outliers are multivariate outliers that differ from conventional outliers because the anomalous values occur for only one feature (or a relatively small number of features).

In [1] the authors present a literature review of various versions of the LOF algorithm in static and stream environments. It collects and categorizes existing local outlier detection algorithms and analyzes their characteristics. It also discusses the advantages and disadvantages of those algorithms and proposes several promising directions for developing improved local outlier detection methods for data streams. Wang et al. [14] provided the progress of outlier detection algorithms until 2019 and illustrated the different outlier detection methods. The paper presents theoretical aspects of outlier detection algorithms, including evaluation techniques and tools for outlier detection.

All the works mentioned here relate to detecting outliers in large data sets or relating to qualitative data. What distinguishes our work from those mentioned above is the analysis of complex data, which is the rule-based representation of knowledge. We are not familiar with similar works.

1.2. Structure of the article

The structure of the article is following. Rule clustering and cluster quality measuring is the subject of Section 2. In Section 3 we present the introduction to outlier detection algorithms with specification of LOF algorithm. Section 4.1 explains in details the two-phase process of discovering outliers in rule-based knowledge bases while Section research describes the experimental part of our research with the analysis of the experiments results.

2. Clustering and outlier mining in rules

Clustering is one of the unsupervised learning methods. To find the optimal group structure, we need to use cluster quality measures. When the knowledge base contains unusual rules, it may not be possible to obtain a good quality rule cluster structure. In the literature, several measures of cluster quality assessment have been proposed, which allow checking whether the obtained structure is optimal. In our research, we hope that when we select unusual rules, the quality of clusters composed of similar rules will improve. For this purpose, we will use such cluster quality assessment measures twice: after clustering but before exploring outliers in the rules and then after removing unusual rules. The marked improvement in cluster quality confirms that the rules were outlier and should not be clustered with others. On the other hand, domain experts should subject them to further analysis and perhaps ensure the completeness of the domain knowledge in this area, i.e., for example, adding new rules.

In our research, we analyze rule-based knowledge bases with real data, and we try to develop the methods for efficient exploration of such rules. Rules in the form $IF - THEN$ are very popular methods for the representation of experts' knowledge. No matter who uses the knowledge, physicians or economists, rules can be convenient for various domains. Rules are probably the most natural way of explaining the way of domain experts' thinking. Rules can create chains so that the conclusion of one rule can then be a premise in another one. When the rules form such chains in the knowledge base, their analysis can be complicated. The real nature of this type of data means that unusual need not be false rules. It is enough that they relate to a rarely explored area of domain knowledge and will be significantly different from the rest of the rules. Finding them in the set of all rules can allow a knowledge engineer to take care of the completeness of the knowledge base and successfully implemented the inference process. What is more, most often, rules (their premises and conclusions) are created using various attributes: quantitative, qualitative, binary. In addition, some rules may be concise (containing several premises) while others may contain multiple premises. All this makes the rules very unusual data structures and require specialized analysis.

Efficient exploration of rules requires reorganization of them in the structure of rule clusters. To do that, we need to use a proper clustering algorithm. After analyzing various clustering methods, we finally choose the hierarchical algorithm *AHC* (Agglomerative Hierarchical Clustering). This technique is more natural (than degglomerative) when we want to cluster the data until they are similar enough. Due to the fact that rules subject to clustering are complex data, and often contain not only quantitative but also qualitative attributes, we decided to use the Gower measure as a measure of the distance between the grouped data.

In papers [8], and [7], we devoted attention to describing the *AHC* algorithm and its application to such specific data as rules in knowledge bases [5]. It's worth recalling that the clustering flow can be different depending on the clustering method we used. Hence, in our research, we use several options [1] so that we can always find the optimal solution [9].

When the knowledge base contains unusual rules, it is difficult to build clusters with high quality. Such rules influence the quality of created clusters, and we decided to examine this in detail.

In real knowledge bases with rules, there may be outliers in the data. The detection of such unusual rules may speed up the clustering process and improve the efficiency of inference by the better formation of clusters. To check which rules are an outlier and have a negative impact on the quality of rule clusters, we suggest using the *LOF* algorithm and measuring the quality of the clusters before detecting unusual rules and after their detection. Cluster quality indexes should improve after selecting outliers.

To examine whether we built rule clusters with good quality, we need to measure their quality using one of the following indices. The most popular measures of cluster quality assessment [15] are presented in Table 1.

3. Outlier detection

Outlier detection is the term used for anomaly detection, fraud detection, and novelty detection. The purpose of outlier detection is to detect rare events or unusual activities that differ from most data points in a dataset. Recently,

[1] single linkage, complete linkage, average linkage, McQuitty, UPGMA, and Ward's method

Table 1. Cluster quality measures

name	equation
Dunn	$D = min_{1\le i\le c}\{min_{1\le j\le c, j\neq i}\{\frac{\delta(X_i,X_j)}{max_{1\le k\le c}\{\Delta(X_k)\}}\}\}$ $\delta(X_i, X_j)$ is the inter-cluster distance between cluster C_i and C_j, and $\Delta(X_k)$ is the intra-cluster distance of cluster X_k.
Davies-Bouldin	$DB = \frac{1}{k}\sum_{i=1}^{k} max_{i\neq j}\{\frac{\Delta(X_i)+\Delta(X_j)}{\delta(X_i,X_j)}\}$ for k number of clusters.
CPCC	$CPCC = \frac{\frac{1}{M}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} d_{ij}\cdot c_{ij} - \mu_p\cdot\mu_c}{\sqrt{((\frac{1}{M})\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} d_{ij}^2 - \mu_p^2)((\frac{1}{M})\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} c_{ij}^2 - \mu_c^2)}}$ where $\mu_p = \frac{2}{N(N-1)}\sum_{i=1}^{N}\sum_{j=i+1}^{N} d_{ij}$ and $\mu_c = \frac{2}{N(N-1)}\sum_{i=1}^{N}\sum_{j=i+1}^{N} c_{ij}$, d_{ij} and c_{ij} be the (i, j) element of P and P_c, respectively for cophenetic matrix P_c and the proximity matrix P of X with $M = \frac{N(N-1)}{2}$
Sillhouette	$s(i) = \frac{b(i)-a(i)}{max\{a(i),b(i)\}}$ for data point $i \in C$, $a(i) = \frac{1}{\lvert C_i\rvert - 1}\sum_{j\in C, j\neq i} d(i, j)$ being the mean distance between i and all other data points in the same cluster, $d(i, j)$ being the distance between data points i and j in the cluster C_i, whereas $b(i) = min_{k\neq i}\frac{1}{\lvert C_k\rvert}\sum_{j\in C_k} d(i, j)$ being the smallest (hence the *min* operator in the formula) mean distance of i to all points in any other cluster, of which i is not a member.

outlier detection has become an important problem in many applications such as in health care, fraud transaction detection for credit cards, and intrusion detection in computer networks.

3.1. Outlier's definition

In machine learning, the detection of „not-normal" instances within datasets has always been of great interest. This process is commonly known as *anomaly detection* or *outlier detection*. According to the definition given by *Grubbs* in 1969: „An outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs" [3].

Classic outlier detection algorithms most often use distance measures for quantitative types of data. Such algorithms assume that all data have the same data representation space. In the case of rule representation, we can not make such an assumption. The real knowledge base has got rules with different structures (different lengths and data types). We assume that a given rule is a kind of outlier if its similarity to the others is too small. In the literature, there are various similarity measures suitable for rule representation. Correct analysis of the similarity between the rules requires that before starting the outlier mining process to transform all rules into a structure in which each rule will be a vector of a fixed length equal to the number of attributes forming the premises. Only indexes corresponding to the attributes that make up a given rule will be analyzed. In the literature on outlier detection algorithms, one of the most popular ones is the *LOF* algorithm, described below. Unusual rules are not created as the result of an error but have got an unusual feature and, in that context, differ from other rules in a given knowledge base. Such rules have to be discovered by knowledge engineers and discussed with knowledge experts. Knowledge experts knowing the unusual rules in some context can extend the domain to which an unusual rule belongs. In other words, the fact that we are unable to discover the unusual rules in some context brings the opportunity to extend the domain knowledge. For domains such as medicine, economy, etc., it is a significant issue.

3.2. Analysis of the LOF algorithm

In anomaly detection, the *LOF* (Local Outlier Factor) algorithm is an algorithm proposed by Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander in 2000 for finding anomalous data points by measuring the local deviation of a given data point concerning its neighbors [6]. As the name of the algorithm suggests, the *LOF* measures the local deviation of a data point $p \in D$ concerning its k nearest neighbors. A point p is declared anomalous if it's *LOF* is large. The *LOF* factor is based on a concept of the local density, where locality is given by k nearest neighbors, whose distance is used to estimate the density. By comparing the local density of an object to the local densities of its neighbors, one can identify regions of similar density and points that have a substantially lower density than their neighbors. These are considered to be outliers.

Local density based methods compare the local density of the object to that of its neighbors. Imagine that we are looking for outliers in rule-based knowledge base. The *LOF* of a given rule is obtained as described in the following steps:

1. Find the distance, $d_k(p)$, between p and its k-th nearest neighbor. The distance can be any measure, but typically the Euclidean distance is used.
2. Let the set of k nearest neighbors of p be denoted by $N_k(p) = \{q \in D - \{p\} : d(p,q) \leq d_k(p)\}$.
3. Define the reachability distance of a rule q from p, as $d_{reach}(p,q) = max\{d_k(q), d(p,q)\}$. It strongly depends on the value of k, because we take into account only the k-nearest neighbors of a given rule.
4. The average reachability distance of p is $\overline{d_{reach}}(p) = \frac{\sum_{q \in N(p)} d_{reach}(p,q)}{|N_k(p)|}$ The local reachability density of a rule is defined as the reciprocal of reachability distance $l_k(p) = \frac{1}{\overline{d_{reach}}(p)}$.
5. Finally, this local reachability density is compared with the local reachability densities of all rules in $N_k(p)$, and the ratio is defined as *LOF* (Local Outlier Factor): $L_k(p) = \frac{\sum_{o \in N_k(p)} \frac{l_k(o)}{l_k(p)}}{|N_k(p)|}$
6. The *LOF* of each rule is calculated, and rules are sorted in decreasing order of $L_k(p)$. If the *LOF* values are large, the corresponding rules are declared as outliers.
7. To account for k, the final decision is taken as follows: $L_k(p)$ is calculated for selected values of k in a pre-specified range, max $L_k(p)$ is retained, and a p with large *LOF* is declared an outlier.

LOF compares the density of any given data point to the density of its neighbors. Since outliers come from low-density areas, the ratio will be higher for anomalous data points. As a rule of thumb, a normal data point has a *LOF* between 1 and 1.5 whereas anomalous observations will have much higher *LOF*. The higher the *LOF* the more likely it is an outlier.

4. Experiments

The Section presents both the research methodology, the characteristics of the source data, and the results of the experiments and their analysis.

4.1. Methodology of our research

Four different cluster quality assessment measures (Dunn, Davies-Bouldin, Sillhouette, CPCC) will allow us to choose a structure in which the number of rule clusters maintains optimal rule cluster consistency and separation values. In the second stage, we look for outlier rules, generating 1%, 5%, and 10% of the rules that are the most out of the rest, taking into account the conditional part of the rules, using the *LOF* algorithm. Then we discard the rules indicated as an outlier from the set (in fact, we exclude them into a separate set and return them to the domain expert to work on the exploration of these rules and the domain knowledge related to them) and re-evaluate the quality of the clusters of rules - using the 4 above-mentioned measures of cluster quality but without any unusual rules. We assume that the aforementioned measures should be improved, which will confirm that the designated rules were outlier and disrupt the rule clusters' structure. This is detailed below.

1. Phase I - rules induction and clustering:
   (a) Loading source dataset into RapidMiner Studio [2] and Sampling using RapidMiner Studio,
   (b) Loading the source dataset into RSES [3],
   (c) Generating rules with RSES using $LEM2$ algorithm,
   (d) Rule clustering: for each of the number of k groups from 2 to $\sqrt{number\_of\_rules}$ perform AHC hierarchical clustering, repeat changing k groups and selecting one of the clustering methods (single, complete, average,

[2] https://rapidminer.com/
[3] https://www.mimuw.edu.pl/~szczuka/rses/

centroid, mcquitty, median, ward.D and ward.D2). Designate four metrics (silhouette index, Dunn index, Davies-Bouldin index and CPCC cophenetic correlation coefficient) and find the maximum/minimum value of as many metrics as possible and indicate the optimal result, i.e. the optimal number of groups to trim the dendrogram after the *AHC* clustering and the optimal clustering method.

2. Phase II - in our research, we want to learn about unusual rules, which constitute 1%, 5%, 10% of all rules in the knowledge base, using searching for LOF outliers.

   (a) Detecting unusual rules using *LOF* algorithm. Search for deviations by selecting a *k*-distance from 2 to $\sqrt{number\_of\_rules}$, until you reach the $\sqrt{number\_of\_rules}$.
   (b) Removing 1% of outlier rules from the knowledge base (refer custom rules to the field expert for analysis) and return the clustering process. Repeat the deletion and grouping for 5% and 10% of detected custom rules,
   (c) Recalculating all four grouping quality indicators (silhouette index, Dunn index, Davies-Bouldin index and CPCC cophenetic correlation coefficient),
   (d) Verifying how many cases the quality of the cluster improved, in how many cases the indicators showed a deterioration in the quality of the newly formed groups, and in how many cases remained unchanged and did not react to the deletion of non-standard rules,
   (e) Analysing the results of the studies and evaluate the selected indicators for consistency of results. Compare the differences in outliers discovered using *LOF* algorithm for 1%, 5%, and 10% outliers.

4.2. Data source

Since the beginning of the *Covid* 19 pandemic, the number of scientists working on analyzing this virus has been growing at a remarkable pace. In our research, we wanted to reflect on the effectiveness of detecting outliers in this type of data—so unusual cases. We looked at many free real data repositories and finally picked one of them, an extensive data set: *Covid* 19 Case Surveillance public use dataset containing 8 405 709 observations and have qualitative characteristics [4]. The collection is available on `kaggle.com`[5] Under license CC0: Public Domain[6]. The *Covid* 19 case surveillance system database includes individual-level data reported to U.S. states and autonomous reporting entities, including New York City and the District of Columbia (D.C.), as well as U.S. territories and states. On April 5, 2020, *Covid* 19 was added to the Nationally Notifiable Condition List and classified as "immediately notifiable, urgent (within 24 hours)" by a Council of State and Territorial Epidemiologists (CSTE) Interim Position Statement (Interim-20-ID-01). CSTE updated the position statement on August 5, 2020, to clarify antigen detection tests and serologic test results within the case classification. The statement also recommended that all states and territories enact laws to make *Covid* 19 reportable in their jurisdiction and that jurisdictions conducting surveillance should submit case notifications to CDC. *Covid*19 case surveillance data[7] are collected by jurisdictions and shared voluntarily with CDC. The de-identified data in the public use dataset include demographic characteristics, exposure history, disease severity indicators and outcomes, clinical data, laboratory diagnostic test results, and comorbidities. All data elements there is the *Covid* 19 case report form located at `www.cdc.gov/coronavirus/2019-ncov/downloads/pui-form.pdf`. Table 2 presents a set of 11 qualitative attributes, where 8 attributes are text data (String), and 3 attributes are dates (DataTime).

An attribute *death_yn* is used as the decision attribute. Unfortunately, standard algorithms for generating rules from data, including algorithms (including *LEM*2 [4]) available under the *RSES* [18] tool, would not be able to load such

[4] https://www.kaggle.com/arashnic/covid19-case-surveillance-public-use-dataset
[5] Kaggle, a subsidiary of Google LLC, is an online community of data scientists and machine learning practitioners. Kaggle allows users to find and publish data sets, explore and build models in a web-based data-science environment, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges.
[6] https://creativecommons.org/publicdomain/zero/1.0/
[7] https://data.cdc.gov/Case-Surveillance/COVID-19-Case-Surveillance-Public-Use-Data/

Table 2. Data source description

Attribute's name	Meaning and values
cdc_report_dt	(Date cdc reported)
pos_spec_dt	(Date of first positive specimen collection (MM/DD/YYYY)),
onset_dt	(What was the onset date?),
current_status	(What is the current status of this person?), takes the values: (Laboratory-confirmed case 94%, Probable Case 6%),
sex	(Gender), takes the values: (Female 52%, Male 47%, Other 1%),
age_group	(Age group categorie), takes the values: (20 – 29 Years 19%, 30 – 39 Years 18%, Other 64%),
Race and ethnicity	(combined) (Case Demographic), takes the values: (Unknown 33%, White 31%, Non-Hispanic, Other 36%),
hosp_yn	(Was the patient hospitalised?), takes the values: (No 42%, Missing 38%, Other 20%),
icu_yn	(Was the patient admitted to an intensive care unit (ICU)?), takes the values: (Missing 74%, Unknown 15%, Other 11%),
death_yn	(Did the patient die as a result of this illness?), takes the values: (No 44%, Missing 41%, Other 15%),
medcond_yn	(Did they have any underlying medical conditions and/or risk behaviors?), takes the values: (Missing 72%, Unknown 10%, Other 18%).

a large set of data. We were forced to sample the data using RapidMiner Studio [8]. 0.5% of records were sampled from more than 8 million units (8 405 079), i.e., approximately 4202 cases. After the rules have been generated, 2491 rules have been created in RSES. The largest number of premises in the rules is 10 premises, we have 408 such rules, and the smallest is one rule with 4 reasons. In addition, we have 1046 rules with 9 reasons, 592 usually with 8 reasons, 342 rules with 7 reasons, 102 rules with 6 reasons, and 6 rules with 5 reasons. On average, we have 8.52 reasons for the rule. The largest support for the rule in the rules set is 59. Thus, the base of 2 491 rules will ultimately be subject to hierarchical clustering to optimize the inference processes. At the same time, assuming that there will be outliers in such a rule base, we want to use the *LOF* algorithm to select them.

4.3. Results

In our assumptions, we want to select outlier rules being respectively 1, 5, and 10% of all rules in the knowledge base. 1% of outliers means that we will look for the 25 most unusual rules, 5% respectively 125 unusual rules, and 10% means as many as 250 unusual rules. Of course, it is ultimately the knowledge engineer or domain expert that would decide how many non-standard rules in the knowledge base they want to select and explore further. The *LOF* algorithm is used in order to discover such rules. We want to check if cluster quality measures work well for specific data such as rules and rule clusters. So we want to examine if they respond appropriately to the appearance of outliers in the rule clusters. These indices improve when we eliminate outliers from the cluster because the clusters take on better quality, consistency.

In work [9], we examined 7 different measures of cluster quality assessment. They included both measures, which, while improving the quality of clusters, reduce their baseline value and act inversely. This time, the research included only those measures that assess the quality of clusters in a similar way, and therefore expect an increase in the value of a given index to improve the consistency of clusters by eliminating atypical rules.

We conduct our research on a set of nearly 2500 rules (2491 to be exact rules) generated automatically using the *LEM*2 algorithm as part of the *RSES* package from the *Covid* 19 case surveillance data source. Initially, this set consisted of almost 8.5 million data that the above-mentioned *LEM*2 algorithm is unable to process. Data sampling

[8] Operator Sample creates a sample from an ExampleSet by selecting examples randomly. Details: https://docs.rapidminer.com/latest/studio/operators/blending/examples/sampling/sample.html

was essential. From a sample of 4 202 objects, these 2491 rules were automatically generated. The *AHC* algorithm groups these rules.

Table 3 shows a repetition for 49 different values of the number of groups and 8 different methods of combining the values of 4 selected measures of cluster quality assessment: Dunn, Davies-Bouldin indices, body measures and CPCC measures.

Table 3. Phase 1 - looking for the optimal values of clustering quality

id	Silhouette	Dunn	Davies-Bouldin	CPCC	Clusters	Clustering method
1	0.256469	0.1	1.255498	0.627978	2	Ward.D
2	0.133531	0.1	1.817726	0.627978	3	Ward.D
...						
48	0.123899	0.111111	2.029511	0.627978	49	Ward.D
49	0.124642	0.111111	2.017838	0.627978	50	Ward.D
1	0.149606	0.1	1.875984	0.518124	2	Ward.D2
2	0.189705	0.1	1.540459	0.518124	3	Ward.D2
...						
48	0.151667	0.125	2.104539	0.518124	49	Ward.D2
49	0.147501	0.125	2.099490	0.518124	50	Ward.D2
1	0.255129	0.4	0.577142	0.431075	2	single
2	0.224813	0.4	0.541113	0.431075	3	single
...						
48	-0.230618	0.2	0.788247	0.431075	49	single
49	-0.230997	0.2	0.794906	0.431075	50	single
1	0.194951	0.1	1.376539	0.534238	2	complete
2	0.079259	0.1	2.262753	0.534238	3	complete
...						
48	0.039924	0.166667	2.030654	0.534238	49	complete
49	0.047490	0.166667	2.040525	0.534238	50	complete
1	0.283532	0.2	1.018698	0.747380	2	mcquitty
2	0.226740	0.2	1.488808	0.747380	3	mcquitty
...						
	0.056885	0.125	1.774643	0.747380	49	mcquitty
	0.058056	0.125	1.833024	0.747380	50	mcquitty
1	0.251177	0.2	0.518758	0.518853	2	median
2	0.158096	0.2	0.530860	0.518853	3	median
...						
	-0.288508	0.1	1.209505	0.518853	49	median
	-0.295951	0.1	1.198937	0.518853	50	median
1	**0.312414**	**0.4**	**0.483487**	**0.604610**	2	centroid
2	0.208589	0.2	0.512787	0.604610	3	centroid
...						
	-0.273642	0.2	0.895063	0.604610	49	centroid
	-0.275366	0.2	0.891933	0.604610	50	centroid

It can be seen that the optimal index values were obtained for 2 clusters and the centroid method. Table 4 shows the results of the second phase, in which we are looking for outliers in the data in the optimal structure of rule groups established in the first phase. For this purpose, we choose 1%, 5%, and 10% of the most unusual rules using the

LOF method. We are looking for cases that, after eliminating unusual rules, will improve the evaluation of the cluster quality.

Table 4. Phase 2 - lists of detected outliers for 1%, 5% and 10% case

	LOF outliers id
1%	**359**, 519, 190, 322, 2394, 177, 231, 2356, 276, 218, 196, 948, 2302, 176, 261, 337, 1866, 1015, 2440, 766, 353, 1833, 1902, 362, 2279
5%	2489, 485, 165, 242, 504, 105, 479, 471, 483, 2419, 102, 103, 243, 79, 475, 377, 128, 52, 129, 2173, 2364, 164, 537, 80, 2135, 2358, 51, 172, 24, 100, 63, 244, 2347, **359**, 2377, 252, 40, 2176, 2367, 499, 2155, 2156, 2154, 1330, 1251, 2487, 1354, 1329, 2344, 512, 484, 2411, 202, 2465, 2475, 1208, 526, 251, 246, 358, 2325, 2368, 2164, 2157, 2159, 2175, 468, 1209, 2212, 362, 2393, 2151, 2491, 352, 152, 2482, 173, 2404, 1116, 1152, 71, 1346, 490, 203, 521, 2392, 2455, 1244, 167, 2275, 1299, 346, 1118, 2328, 1312, 1131, 1227, 1194, 1141, 525, 2149, 2169, 1243, 133, 498, 2178, 488, 2397, 1185, 727, 1305, 1262, 126, 354, 1275, 85, 1231, 1229, 1335, 1268, 1255, 1267, 106, 469, 1170
10%	2231, **359**, 82, 2167, 1017, 1358, 519, 2160, 231, 2105, 212, 1270, 2188, 1144, 461, 2394, 85, 70, 1206, 136, 2134, 190, 189, 332, 1337, 350, 218, 33, 346, 2425, 171, 25, 340, 2177, 134, 2033, 40, 322, 2224, 480, 192, 337, 1013, 2120, 221, 1369, 108, 250, 42, 8, 2393, 1946, 1914, 1204, 1322, 196, 948, 5, 2228, 1071, 2423, 2021, 6, 2166, 1290, 1242, 90, 1366, 525, 86, 2029, 456, 1010, 1166, 425, 18, 1356, 304, 524, 138, 2101, 1115, 1288, 109, 2041, 1849, 591, 1314, 2119, 2366, 78, 1162, 179, 1263, 32, 254, 1937, 536, 2410, 21, 642, 2110, 1866, 826, 1153, 1364, 2418, 295, 2093, 881, 919, 903, 819, 529, 947, 936, 284, 143, 713, 1150, 1015, 1075, 1916, 1961, 2133, 2323, 1833, 1902, 1847, 2190, 1229, 1268, 2072, 2014, 1345, 2331, 1734, 1671, 1516, 2100, 916, 1977, 2010, 2053, 407, 2126, 1714, 1703, 463, 841, 865, 2036, 2031, 1921, 549, 1825, 1851, 1874, 1048, 481, 1363, 1987, 1378, 2027, 806, 1442, 1417, 2091, 1568, 370, 1462, 2248, 2270, 2303, 1544, 450, 438, 1821, 1892, 663, 733, 1470, 620, 652, 1505, 2244, 1643, 651, 1373, 394, 1912, 398, 490, 1396, 554, 1498, 1635, 215, 558, 1447, 1283, 1333, 1618, 1404, 1596, 1398, 382, 2311, 77, 1604, 771, 548, 556, 801, 1523, 1482, 2443, 57, 477, 1611, 1627, 1674, 1606, 1381, 2173, 419, 1009, 1471, 584, 623, 632, 1426, 1571, 193, 300, 2489, 121, 547, 567, 587, 624, 846, 1474, 837, 1901, 391, 2312, 1936, 1481, 1070

The most unusual rule, indicated by all trials, is Rule No. 359:

```
(current_status=Laboratory_confirmed case)&(sex=Male)&(icu_yn=Unknown)&(medcond_yn=Unknown)
&(onset_dt=1900/01/01)&(hosp_yn=Unknown)&(Race_and_ethnicity_combined=Unknown)
&(age_group=20 _ 29 Years)&(pos_spec_dt=1900/01/01)=>(death_yn=No[2]) 2
```

When we look at it closely, it becomes clear why it is unusual. Apart from indicating the age group and gender, all other features (attributes) are undefined (value unknown). And just such rules should be indicated to a domain expert or knowledge engineer to assess whether such a rule should be taken into account in the inference processes at all, or rather indicated for further exploration.

The more outliers we detect, the greater is the coverage of outliers. For 1% there are 2 common rules, for 5% 11 rules and for 10% 14 rules. The more rules we have in the knowledge base and the more unusual rules we want to determine from such a set, the more difficult it is to interpret the results. It is always the additional parameter of the neighborhood k that tells us how much the rule must deviate in the description from other rules in the set to be considered a deviation.

Table 5 confirms that in each case, with 1%, 5%, and 10% of the outliers detected, the cluster quality assessment indicators improved after removing them from the dataset.

5. Summary

We applied the *LOF* algorithm to complex data, such as rules in knowledge bases. In the description of the rules to a large extent, apart from quantitative features, there are qualitative features that mean that we could not be sure of the effectiveness of the algorithm used. The rules found as a result are actually unusual (consultation with a domain

Table 5. Compare the quality of clusters with outlier and after their removal

	Phase I - before removing outliers				Phase II - after removing outliers				
	Silhouette	Dunn	Davies-Bouldin	CPCC	k	Silhouette	Dunn	Davies-Bouldin	CPCC
1%	0.316253	0.4	0.470753	0.607284	7	**0.312414**	0.4	0.483487	**0.604610**
5%	0.320450	0.4	0.475842	0.635482	300	**0.312414**	0.4	0.483487	**0.604610**
10%	0.325561	0.47	0.479455	0.611951	3	**0.312414**	**0.4**	0.483487	**0.604610**

expert confirms that all the rules indicated as deviations are in some sense atypical). When using this algorithm for real data, we have to execute it many times, changing the parameters of the algorithm. In our case, the parameter to be changed is the degree of the neighborhood, which is a determinant of whether a given rule belongs to a group or is already so distant that it should not be included in this group. Hence, we performed many repetitions because for $k = 1 \ldots 50$ with step 1 and then every 50 up to the value of 2450, so as not to exceed the value of N (2491 rules). The detected outliers were indeed unusual rules because, after their removal, it was possible to improve almost all 4 measures of cluster quality assessment. Future research will focus on two issues: research on the improvement of the effectiveness of inference process in rule-based knowledge bases after the elimination of outlier rules and further search for algorithms that will detect unusual rules even more effectively.

References

[1] Alghushairy, O., Alsini, R., Soule, T. and Ma, X., (2021) "A Review of Local Outlier Factor Algorithms for Outlier Detection in Big Data Streams", *Big Data and Cognitive Computing*, **5**, Nr. 1, https://doi.org/10.3390/bdcc5010001

[2] Goldstein M., Uchida S., (2016) "A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data", *PLoS One*, **2016**, 11(4):e0152173, https://doi.org/10.1371/journal.pone.0152173.

[3] Grubbs, F.E., (1969) "Procedures for Detecting Outlying Observations in Samples", *Technometrics*, **11**, (1):1–21, https://doi.org/10.1080/00401706.1969.10490657.

[4] Grzymała-Busse J.W., (1997) "A new version of the rule induction system LERS", *Fundam. Inform.* **31 (1)** 27-39. https://doi.org/10.3233/FI-1997-3113.

[5] Legendre, P., Fionn M., (2014) "Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion?" *Journal of Classification* **31**: 274-295.

[6] Breunig M. M., Kriegel H., Ng R.T., and Sander, J., (2000) "LOF: Identifying Density-Based Local Outliers". *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, : 93–104, Dallas, Texas, USA.

[7] Nowak-Brzezińska, A., Wakulicz-Deja, A., (2019) "Exploration of rule-based knowledge bases: A knowledge engineer's support", *Information Sciences*, **485**: 301-318, Elsevier

[8] Nowak-Brzezińska, A., (2018) "Enhancing the efficiency of a decision support system through the clustering of complex rule-based knowledge bases and modification of the inferencje algorithm", *Complexity*, **2018**, https://doi.org/10.1155/2018/2065491

[9] Nowak-Brzezińska, A. , Horyń C., (2020) "Exploration of Outliers in If-Then Rule-Based Knowledge Bases", *Entropy*, **22**, 10, https://doi.org/10.3390/e22101096.

[10] Pijnenburg, M., Kowalczyk, W. (2018) Singular Outliers: Finding Common Observations with an Uncommon Feature. Information Processing and Management of Uncertainty in Knowledge-Based Systems. Applications. IPMU 2018. Communications in Computer and Information Science, vol 855. Springer, Cham. https://doi.org/10.1007/978-3-319-91479-4_41

[11] Ranga Suri N.N.R., Murty M N., Athithan G., (2019) "Outlier Detection in Categorical Data". *Outlier Detection: Techniques and Applications*. Intelligent Systems Reference Library, **155**. Springer, Cham. https://doi.org/10.1007/978-3-030-05127-3_5

[12] Sinha A., Jana P.K., (2018) "Efficient Algorithms for Local Density Based Anomaly Detection". Distributed Computing and Internet Technology. ICDCIT 2018. LNCS, **10722**. Springer, Cham. https://doi.org/10.1007/978-3-319-72344-0_30

[13] Thudumu, S., Branch, P., Jin, J. et al., (2020) "A comprehensive survey of anomaly detection techniques for high dimensional big data". *J Big Data*, **7**: 42 (2020). https://doi.org/10.1186/s40537-020-00320-x

[14] Wang, H., Bah, M.J., Hammad, M., (2019) "Progress in Outlier Detection Techniques: A. Survey". *IEEE Access 2019*, **7**, 107964–108000, https://doi.org/10.1109/ACCESS.2019.2932769

[15] Wierzchoń, S., Kłopotek, M.A., (2015) "Algorytmy analizy skupień". WNT Warszawa.

[16] https://www.sciencedirect.com/search?qs=covid

[17] https://www.nature.com/search?q=covid

[18] https://www.mimuw.edu.pl/~szczuka/rses/get.html

[19] https://covid19.who.int/