Knowledge-Based Systems 195 (2020) 105659

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

K-Means-based isolation forest☆

Paweł Karczmarek [a,*], Adam Kiersztyn [a], Witold Pedrycz [b,c,d], Ebru Al [e]

[a] Department of Computer Science, Lublin University of Technology, ul. Nadbystrzycka 36B, 20-618, Lublin, Poland
[b] Department of Electrical & Computer Engineering, University of Alberta, Edmonton T6R 2V4 AB, Canada
[c] Department of Electrical and Computer Engineering, King Abdulaziz University, Jeddah, 21589, Saudi Arabia
[d] Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland
[e] Ekol Lojistik Inc., Ekol Caddesi No:2, TR-34935 Sultanbeyli-Istanbul, Turkey

ARTICLE INFO

Article history:
Received 28 October 2019
Received in revised form 27 January 2020
Accepted 12 February 2020
Available online 17 February 2020

Keywords:
Anomaly detection
Isolation forest
K-Means
Search tree
Spatio-temporal datasets

ABSTRACT

The task of anomaly detection in data is one of the main challenges in data science because of the wide plethora of applications and despite a spectrum of available methods. Unfortunately, many of anomaly detection schemes are still imperfect i.e., they are not effective enough or act in a non-intuitive way or they are focused on a specific type of data. In this study, the classical method of Isolation Forest is thoroughly analyzed and augmented by bringing an innovative approach. This is *k*-Means-Based Isolation Forest that allows to build a search tree based on many branches in contrast to the only two considered in the original method. *k*-Means clustering is used to predict the number of divisions on each decision tree node. As supported through experimental studies, the proposed method works effectively for data coming from various application areas including intermodal transport and geographical, spatio-temporal data. In addition, it enables a user to intuitively determine the anomaly score for an individual record of the analyzed dataset. The advantage of the proposed method is that it is able to fit the data at the step of decision tree building. Moreover, it returns more intuitively appealing anomaly score values.

1. Introduction

The task of anomaly detection also known as outlier detection in large data coming from various aspects of real life, in particular, business data, has been one of the most challenging problems because of fast increase of datasets size and their complexities in modern organizations. An anomaly can be defined in various ways. It may be a deviation in amplitude, randomly inserted value, lack of data, data of various types, and implied by many commonly encountered errors [1,2]. The problem can be even more important in the case when multi-dimensional data are considered, in particular, spatio-temporal time series, where one subset of variables is related to the geographical positions, i.e., latitude–longitude pairs of coordinates or other positions such as points on the plots or maps, while the rest of the variables denote one or more time series values, viz. a time-related values (time stamps). Typical examples of this kind of data are demographic, geological and geophysical data or the details of intermodal transport processes.

There are many types of anomalies which pose challenging problems for the scientists and practitioners in the field. There may be incorrect data present in the database, e.g., mistakenly inserted by users. Also the distant points located far from cluster centers could be regarded as anomalies. One can think of anomaly as missing data in the database or a result of erroneous calculations. Anomalies include transport delay times or extremely short vehicle transit times, in extreme cases, negative, unexpected travel trajectories, lack of transport details and characteristics, or even typos inserted by users or dataset administrators, etc. Anomalies in databases can be represented by readings of parameters of malfunctioning devices, e.g., perimetry or server temperatures variations, as well as network anomalies, various fraud attempts, e.g., impersonating someone in electronic banking systems or anomalies in a crowded scenes. These significant areas of applications and resulting problems lead to the increased involvement in the search for outliers in huge databases. Therefore, the literature of the topic is vast and application-oriented. A few main results will be recalled now.

The experts in data mining often list a few methods efficient in outlier detection domain. Isolation forest [3,4] is almost always

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2020.105659.

* Corresponding author.
E-mail addresses: pawel.karczmarek@gmail.com (P. Karczmarek), adam.kiersztyn.pl@gmail.com (A. Kiersztyn), wpedrycz@ualberta.ca (W. Pedrycz), ebru.al@ekol.com (E. Al).

https://doi.org/10.1016/j.knosys.2020.105659

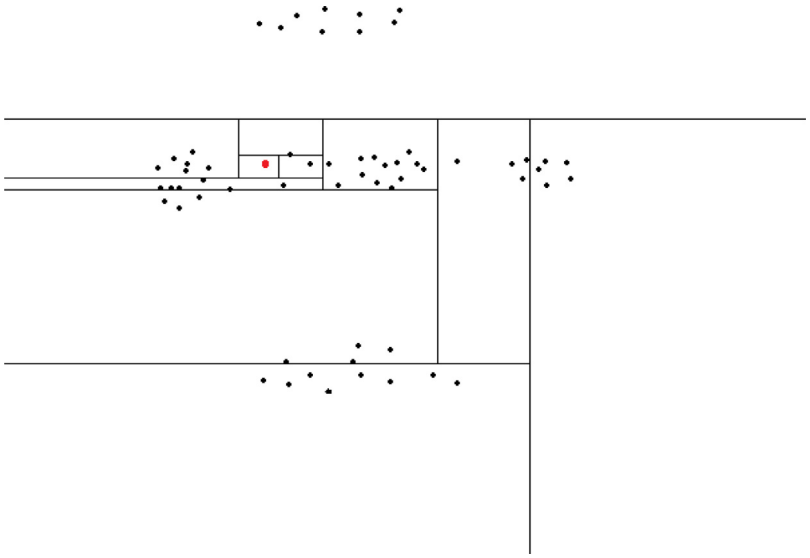

Fig. 1. An intuitive interpretation of isolation forest. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

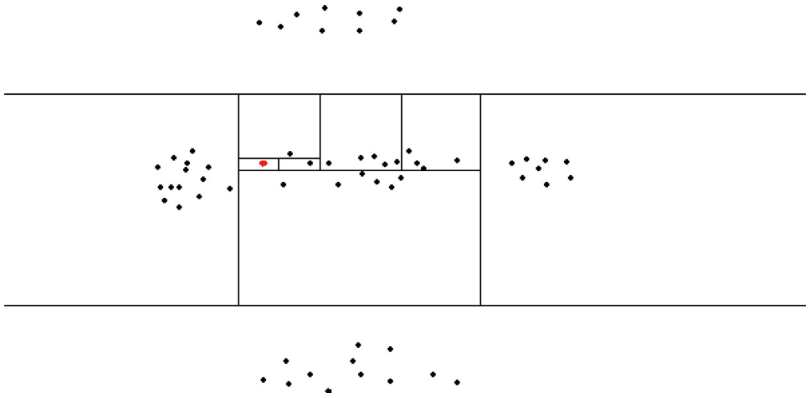

Fig. 2. Clustering of points: Intuition behind the *k*-Means-based IF.

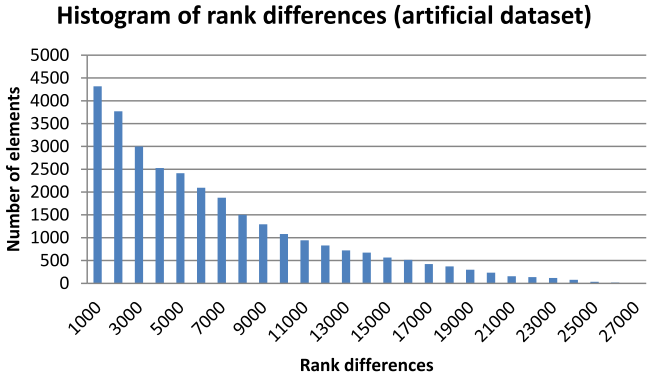

Fig. 3. Rank differences between IF and its *k*-Means-based modification.

one of the methods presented at such rankings. Isolation Forest is based on the binary search trees used to find the partition of multidimensional dataset containing a particular record and estimate its anomaly score using a relatively sophisticated formula.

The main goal of this study is to depart from the classic binary search tree structure and augment the trees by engaging the concept of clusters. Namely, the number of leafs of a tree (to be more precise, a search tree) depends on the optimal number of clusters present in the sub-partition of a dataset at the stage of the training. Moreover, the concept of anomaly score is quantified using the membership value to the cluster. This augmentation of the existing method offers interesting and interpretable outcomes that offer a new deeper insight into the data and detected outliers. Moreover, it seems to be faster for large datasets containing mixed data. In this study, a *k*-Means-based modification of the well-known isolation forest (IF) is presented and the main advantages and shortcomings of the method are shown. The novel structure of the search trees built in the process of training is discussed. Next, a certain anomaly score based on the distance of a record to the clusters representing the tree leafs is introduced. Finally, in a series of experiments we present in detail how the proposal works in case of various types of data, namely geographical, spatio-temporal, and mixed datasets including categorical, spatial as well as time series.

An important advantage of the presented approach comes with its intuitive facet. An intuitive anomaly score calculation process with no need of normalization shows clearly which records in a dataset are deemed suspected to be outliers or anomalies. Moreover, the series of experiments conducted here, shows that the intermodal transport data and taxi transport details containing erroneous values can be better analyzed using our approach than the previous one, i.e., the classic isolation forest. It

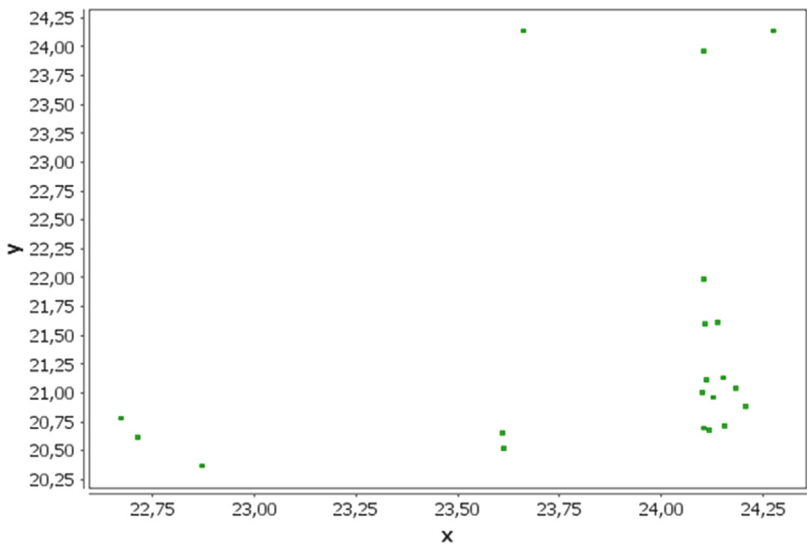

Fig. 4. Points isolated by *k*-Means-based IF but not isolated when using isolation forest.

is worth to stress that despite the fact that the anomaly detection constitutes an intensive area of applications, we limit ourselves to the case of transport data. The motivation is that such kind of data can be either categorical, spatial, spatio-temporal, or mixed. Therefore, it represents a relatively wide spectrum of interest. Finally, it is important to note that the results of numerical experiments prove the intuitiveness of the *k*-Means-based isolation forest. It marks the points as the isolated if they are placed, for instance, outside large groups of other points or if they are boundary values.

The structure of the paper is as follows. In Section 2, the literature of the topic is overviewed. In Section 3, a concept of isolation forest approach is recalled. Moreover, a proposal of its enhancement is presented. In Section 4, the results of numerical experiments with various publically available datasets as well as our private data are detailed. Finally, conclusions and future work directions are presented in Section 5.

2. Overview of previous research

One of the classic approaches to anomaly detection is a density-based technique. It includes *k*-nearest neighbor methods [5–8] and isolation forest [3,4] which is based on a forest of binary search trees trained on samples of a dataset. Another ways of outlier detection are the methods realizing support vector machines [9], core vector machine [10], kernel methods [11], or neural networks, in particular deep learning models such as autoencoders, long-short term memory, or self-organizing maps [12–15]. Also, an interesting approach is based on cluster analysis [16,17], in particular, DBSCAN algorithm [18,19] or DBSCAN and *k*-Means [20], or fuzzy set techniques [21–24]. *k*-Means was also used in a combination with IF, a so-called CBiForest [25], where authors apply it to preselection of dataset records. The dataset is divided into two clusters. The smallest one is considered as abnormal data. Next, the anomaly scores are calculated using the IF method. Another approach falling within this category was an application of Fuzzy *c*-Means, see [26–28]. A very practical method for mining high utility patterns in the case of incremental databases was presented in [29]. Finally, evolutionary Random Weight Networks were applied to detect and identify spam [30]. A particular and important kind of datasets are the databases containing spatial, temporal (time-series) or spatio-temporal data. These large datasets contain information coming from various, often weakly related systems, e.g., transportation databases. To work with such data, the researchers came with various methods. First of them is a set of techniques based on similarity of time series. It may be built on a basis of cross correlation between time series present in a dataset [31]. Another example is the proposal [32], where a nearest neighbor-based distance between so-called discords (the longest time series) is considered, or in [33], where a special dissimilarity measure based on the size of the data was introduced. Outliers for climate changes using distance and neighborhood concepts were discussed in [34]. The second approach is strictly related to the classification task, where classifiers are trained to differentiation among time series with or without anomalies. Various authors proposed neural networks [35], selection inspired by an immune system [36], support vector machines [37], deep neural networks [13,38–40]. Another approach is based on time series modeling. The adopted models include, for instance, autoregressive model [41,42], weighted graph representation [43], hidden Markov model [44], Bayesian network [45,46], scan statistics based on expectations [47], etc. The last set of methods has its origin in clustering, in particular, above-mentioned Fuzzy *c*-Means clustering, FCM, [48]. In general, using this method, one has to assume that time series are clustered with the use of a clustering approach, next the centers of clusters are specified, and the anomaly scoring is determined for particular time series. FCM clustering with a reconstruction criterion to assign the anomaly scores for examined time series related with weather data was applied in [26–28]. *K*-medoid clustering was proposed in [49] while fuzzy *c*-Medoids was proposed in [50]. An interested reader can refer to the in-depth comprehensive surveys of the methods [1,2,51,52].

3. Existing methodology and the proposed generalized approach

The method [3,4] is composed of two general stages, namely: (i) training based on building binary search trees completed on a basis of samples of the dataset and (ii) scoring based on searching these trees when arguments are all the records of the dataset. Let us recall the process of binary trees building. Let D denotes the whole dataset, containing R records, each of the records having Q attributes. Let the number of decision trees be t and the number of randomly chosen sample records used for building each tree be n, i.e., to build the tree it is used the dataset $X \subset D$ containing elements x_i, $i = 1, \ldots, n$.

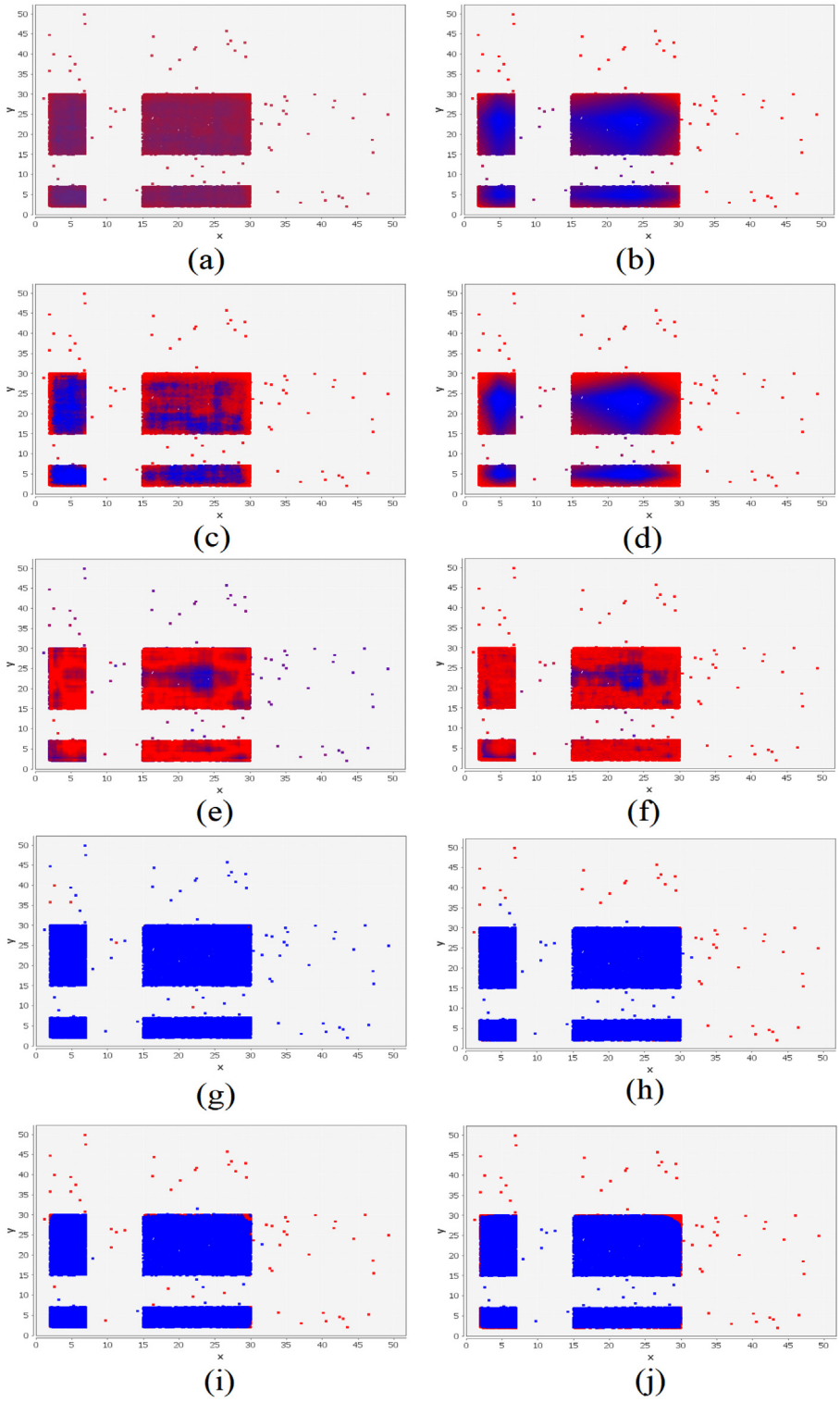

Fig. 5. (a) Results obtained with isolation forest. (b) *k*-Means-based isolation forest. (c) IF results ranked from the less (blue) to the most (red) isolated one. (d) Similar experiment obtained for *k*-Means IF. (e) Module of the difference between normalized scores of IF and *k*-Means IF. (f) Module of the difference between ranked scores of IF and *k*-Means IF. (g) 100 top most isolated points by IF. (h) 100 top most isolated points by *k*-Means IF method. (i) 1000 top most isolated points by IF. (j) 1000 top most isolated points by *k*-Means IF method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Each decision tree is constructed as follows. Using the subsample set$X$, one randomly selects attribute q and then randomly selects its threshold (cutoff) value v. It divides the set of the values of the attribute into two subsets representing two nodes of the root. Next, for each of the subsets, one randomly selects attributes and their appropriate values related to the filter coming from the root. The process is continued until the consecutive subsets are empty or have a single element. Obviously, it is essential to set a maximal tree depth l which is suggested to be $l = \log_2 n$.

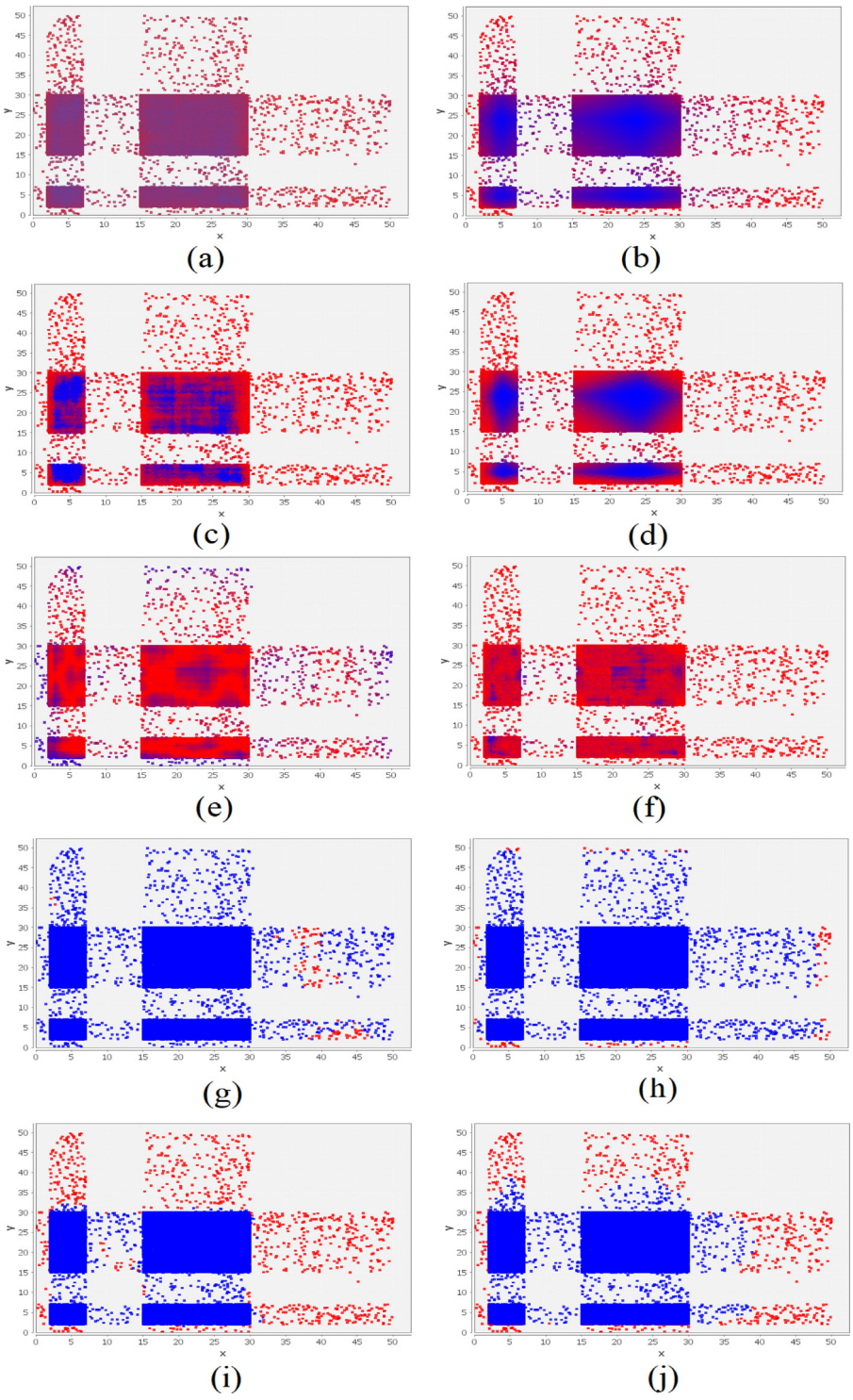

Fig. 6. Plots for the dataset with 50,000 elements. (a) Results obtained with isolation forest. (b) *k*-Means-based isolation forest. (c) IF results ranked from the less (blue) to the most (red) isolated one. (d) Similar experiment obtained for *k*-Means IF. (e) Module of the difference between normalized scores of IF and *k*-Means IF. (f) Module of the difference between ranked scores of IF and *k*-Means IF. (g) 100 top most isolated points by IF. (h) 100 top most isolated points by *k*-Means IF method. (i) 1000 top most isolated points by IF. (j) 1000 top most isolated points by *k*-Means IF method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

At the second stage, the trees are used to determine the scores of anomaly for each point in the dataset. The trees form a forest called isolation forest. For all elements in the original dataset, the anomaly score is calculated by searching each tree. This score value is obtained in the following way [21]:

$$s(x) = 2^{\frac{c(R)}{E(t,M,k)}} \tag{1}$$

where

$$c(R) = 2H(R-1) - 2(R-1)/R \tag{2}$$

is the average path length of binary search tree unsuccessful search process, see [21], while $H(\cdot)$ is estimated as $H(R-1) = \ln(R-1) + 0.5772156649$, R is a number of records in a dataset,

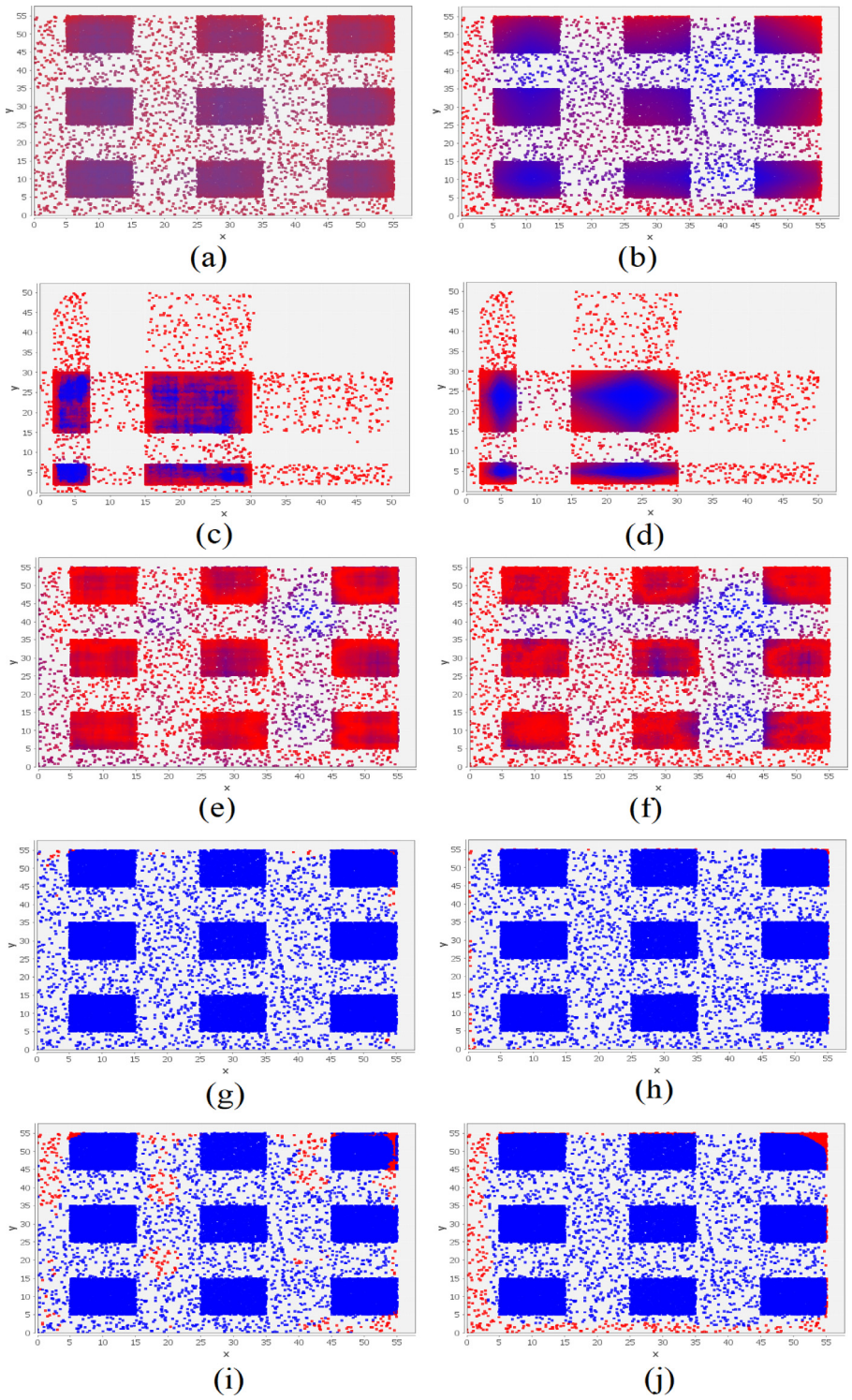

Fig. 7. The results with outputs for the points grouped in 9 geometrical figures. (a) Results obtained with isolation forest. (b) *k*-Means-based isolation forest. (c) IF results ranked from the less (blue) to the most (red) isolated one. (d) Similar experiment obtained for *k*-Means IF. (e) Module of the difference between normalized scores of IF and *k*-Means IF. (f) Module of the difference between ranked scores of IF and *k*-Means IF. (g) 100 top most isolated points by IF. (h) 100 top most isolated points by *k*-Means IF method. (i) 1000 top most isolated points by IF. (j) 1000 top most isolated points by *k*-Means IF method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and finally [6]

$$E(t, M, k) = \frac{1}{t}\sum_{i=1}^{t}\begin{cases}\sum_{j=1}^{M} 1 \text{ if } k = 1\\ \sum_{j=1}^{M} 1 + c(k) \text{ otherwise}\end{cases} \tag{3}$$

Here, t is the number of trees, M is a total number of binary splits completed during search, and k is a cardinality of exit (final) node.

A general interpretation of the above score values is that if the score measure of an instance is close to 1, it is likely anomaly value. If s for some record is much smaller than 0.5 then it can be deemed to be "normal".

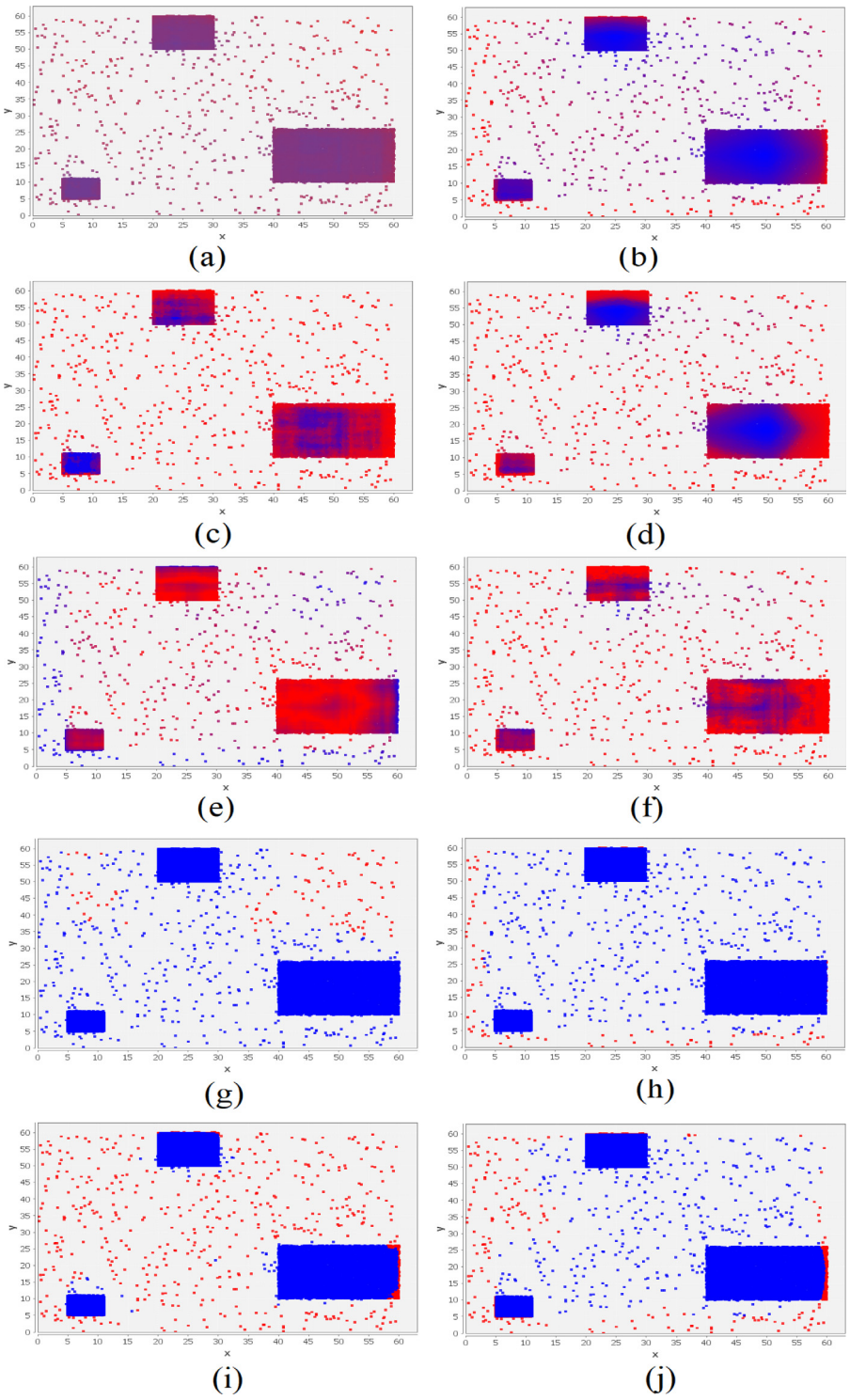

Fig. 8. Three figures and 30,000 points. (a) Results obtained with isolation forest. (b) *k*-Means-based isolation forest. (c) IF results ranked from the less (blue) to the most (red) isolated one. (d) Similar experiment obtained for *k*-Means IF. (e) Module of the difference between normalized scores of IF and *k*-Means IF. (f) Module of the difference between ranked scores of IF and *k*-Means IF. (g) 100 top most isolated points by IF. (h) 100 top most isolated points by *k*-Means IF method. (i) 1000 top most isolated points by IF. (j) 1000 top most isolated points by *k*-Means IF method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

An intuition behind the isolation forest is presented in Fig. 1. Assuming that the dataset is divided alternately according to the x and y values of the Cartesian plane, to find the red point as the singleton in its neighborhood 10 splits are needed. Here, one can note that regardless of whether this point is close to the limits of split area filtered after each step of split operation or not, the final value of splits is 10 which is the input (argument) to the s function. The method is strictly binary with sensitivity to the location of the point related to the division points. Also, note that the method is insensitive to the naturally observable clusters. Therefore, the formula of anomaly scoring is strictly related to the number of splits during binary search of a tree.

Next, an innovative modification of a method described in the previous section is discussed. Let the main symbols introduced

Fig. 9. Top 1000 most isolated pickup points according to *k*-Means-based isolation forest (marked in red). The blue points are the points considered as less isolated. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

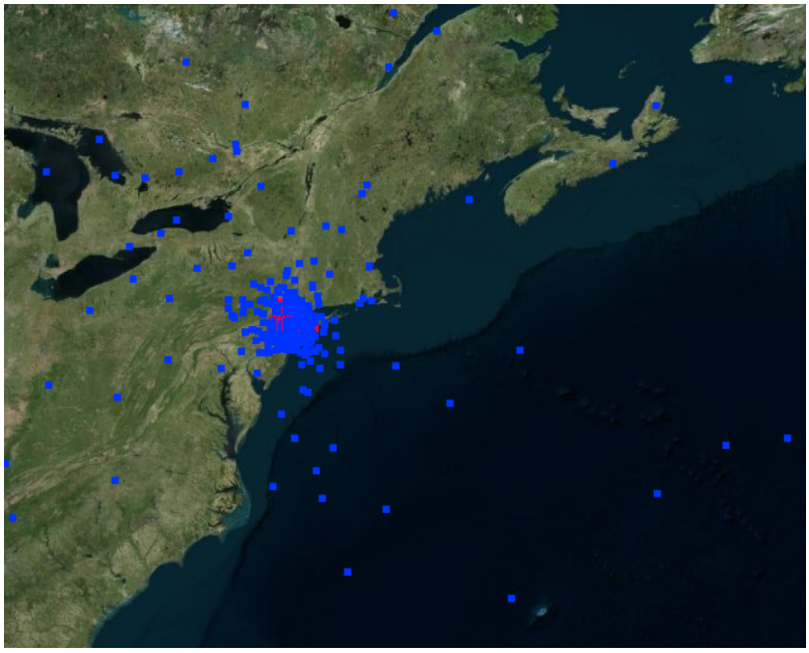

Fig. 10. Top 1000 most isolated pickup points according to isolation forest (marked in red, zoomed image). The blue points are the points considered as less isolated. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

above denote the same artifacts. During the training the process starts similarly as in the previous case. An attribute q is randomly chosen. However, the interval of values between the maximal value of the set X and its minimal value is divided into c clusters. The number c of clusters is selected by applying the elbow rule [53,54]. Namely, assume that c stands for some number of clusters from the range 1, 2, ..., C. The processes of clustering of the values of attribute q, say, k-Means [55] is carried out. When sweeping through the number of clusters, the centers and the limits of clusters are remembered along with the clustering error. The data are the inputs to the elbow method and the optimal number c is selected here. If the error is matched to the number of clusters and one creates the histogram of the values, it often looks like an arm. Therefore an "elbow" point relates to the optimal number of clusters. If the number of clusters is higher than this value, then the clusters do not change the model of data significantly. On the other hand, if the value is small then the sum of clustering errors might be too high and unacceptable. Next, the clusters and their limits create the leafs for the node and for each of these leafs one can repeat the process with the assumption that filter coming from the node is active.

Let us now discuss the process of determining the anomaly score. Each record traces the tree according to its membership to particular cluster. The score value obtained at each split is

$$s = 1 - d\left(x, c_q\right) / d\left(c_l, c_q\right) \tag{4}$$

where c_q is the cluster center and c_l is the cluster limit. s is regarded as the membership of x to the cluster. Final score for each dataset record is its sum of the split memberships (at each jth split, $j = 1,\ 2,\ \ldots,\ M$) divided by the trees number, i.e.,

$$a\left(x\right) = 1 - \frac{1}{t}\sum_{i=1}^{t}\sum_{j=1}^{M} s_j\left(x\right) \tag{5}$$

This approach exhibits an evident advantage, by delivering some intuitive insights into the obtained results. To be precise, the highly isolated records come with the anomaly score close to zero, or even zero, when they do not fit to any cluster. Of course, the records being "typical" bring higher values of the scoring function.

Moreover, at the stage of construction the trees learn the data structure. The number of clusters of chosen argument values, filtered at the earliest levels of tree building, is chosen to fit this structure and show its variety as well as similarity regions. Moreover, it means that the depth of the tree is lower and, therefore, at

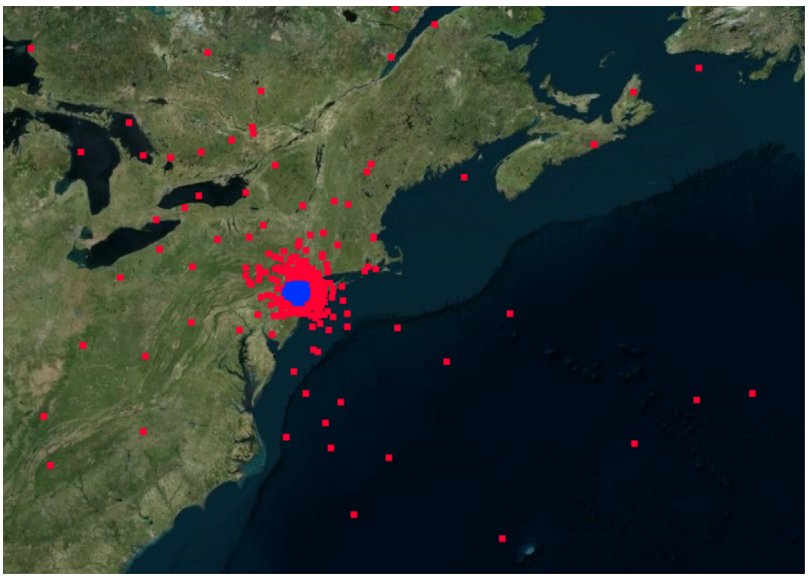

Fig. 11. Top 1000 most isolated pickup points according to k-Means-based isolation forest (marked in red, zoomed image). The blue points are the points considered as less isolated. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the level of searching the time is less. Finally, the number of nodes at each split point is not important from the functionality point of view and computing overhead (see the experimental section) since their number is not high and, in practice, oscillates between 2 and 4 rather than assuming higher values.

The overall process is intuitively presented in Fig. 2. If one starts to divide the points horizontally and then alternatively according to their x and y values, it is possible to note that the red point belongs to the central cluster (relatively near to its center) in the first split. Next, again, it belongs to the second cluster (vertically), higher cluster (horizontally), and to the left cluster (vertically), to the lower cluster (horizontally), and, finally, to the left cluster. The sum of its membership grades is around $3.5 \approx 0.7 + 0.2 + 0.2 + 0.6 + 0.9 + 0.9$, where the particular values are the membership grades to the clusters it belongs to at each stage of splitting. It is noticeable that the values depict its real locations in relation to the cluster centers and are not binary but real numbers. Moreover, the method of division of each part of the tree is more intuitive because it divides the set into clusters in relation to data not totally randomly. From one perspective the trees are "wider", namely the trees may have more nodes at each level. From the second one, they are not deep and the process of searching using recursive procedures can be shortened as the trace does not go as deep as in the isolation forest case. However, more number of comparisons at each node and membership calculations impact the time of the program execution.

4. Numerical experiments

A series of experiments for isolation forest and k-Means isolation forest in-depth analysis using three datasets have been conducted. The first of them is the set of randomly generated points with uniform distribution located in two-dimensional space. Each of the subsets contains 30,000 or 50,000 points. The second one is the publicly available New York City Taxi Trip Data [56] containing records of New York taxi travels including times and geographical coordinates. Finally, we have analyzed the private dataset obtained from the logistic company operating in Europe.

Following the recommendations presented in [3,4], the number of decision trees was set to $t = 100$ and the number of samples $n = 128$ or $n = 256$. In our case it is always used the first of the values to compare both methods, namely isolation forest and k-Means-based isolation forest. It is worth noting that for comparison purposes all the experiments were conducted with the following constraints: All the trees have maximal depth set to 9 and there are always 100 trees used in each experiment.

4.1. Artificial datasets

In the first series of experiments, four kinds of artificial sets containing points grouped in geometrical figures and a few points being located outside these figures have been generated. Figs. no. 1, 3, and 6 present the results of experiments for 30,000 points while Fig. 2 depicts the results for 50,000 artificial points. The plots show that k-Means-based version of isolation forest does not mark the stripes inside figures as isolated. Moreover, it shows that the points located inside the geometrical figures are strictly "normal" and not isolated ones. The plots (b) of all the figures show that the points laying outside the figures but in-between two figures are not "strongly" isolated according to our proposal. However, the plots showing ranks of top 1000 most isolated points (j) show that these coordinates are also high in the hierarchy of isolated points.

From the analysis, it should be noted that the proposed method assigns a low isolation score to points positioned at the center of clusters, while the ratio increases when approaching cluster boundaries. In turn, in the case of the isolation forest method, this distribution is more even. This observation is very well visible in charts (e) and (f) of all the figures presenting differences (Figs. 5–8) in the classification of both methods. The rank difference histogram is shown in Fig. 3. It can be seen that the majority of the distribution mass is focused on small values, however, for up to 3 elements, the rank difference module is over 25,000. This value is close to 30,000 and therefore, it seems to be too large. The twenty points with the largest difference in rank are presented in Fig. 4. These are noise points that the k-Means-based method classified as isolated points, while the IF method made a completely different classification.

4.2. NYC taxi trip data

This dataset is a collection of randomly chosen 3,386,426 records of NYC Taxi Trip Data set in 2013. Each record has

Table 1
Example of data from the NYC Taxi Trip Data.

Medal-lion	Hack license	Vendor id	Rate code	Store and fwd flag	Pickup datetime	Dropoff datetime	Passenger count	Trip time in secs	Trip distance	Pickup longitude	Pickup latitude	Drop-off longitude	Drop-off latitude
0	0	0	1	0	1 956 600	1 957 200	1	600	1.51	−736.5	40.744366	0	0
0	0	0	1	0	1 957 200	1 957 620	1	420	1.43	−736.5	40.744366	0	0
0	0	0	1	0	1 957 680	1 959 000	1	1320	4.81	−736.5	40.744366	0	0
1	1	0	1	0	732 660	733 620	1	960	1.56	−79.822159	42.86293	−79.822159	42.86293
1	1	0	1	0	731 340	732 420	1	1080	4.46	−79.822159	42.86293	−79.822159	42.86293
2	2	0	1	0	2 282 880	2 283 600	2	720	3.63	−78.800003	40.743423	−73.975182	40.770119
2	2	0	1	0	2 283 660	2 284 140	2	480	1.49	−78.800003	40.743423	−73.975182	40.770119
3	3	0	1	0	1 936 200	1 936 620	6	420	1.25	−74.713081	39.992546	−74.713081	39.992546
3	3	0	1	0	1 935 240	1 935 780	6	540	1.36	−74.713081	39.992546	−74.713081	39.992546
4	4	0	1	0	1 003 920	1 004 520	1	600	2.34	−74.67514	41.908611	−74.67514	41.908611

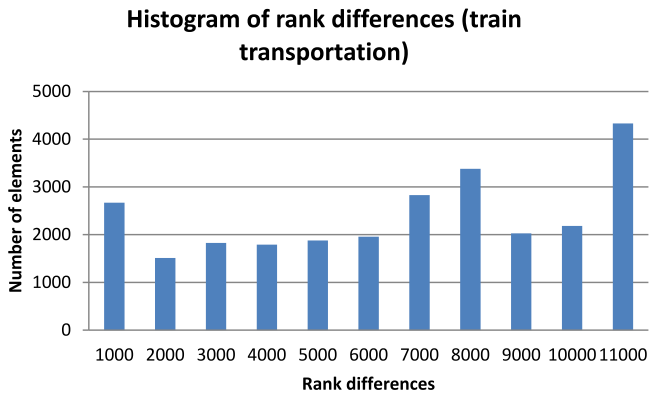

Fig. 12. Rank differences between methods (train transportation).

the following attributes: medallion (anonymized), hack license (anonymized), vendor id, rate code, store and fwd flag, pickup datetime, dropoff datetime, passenger count, trip time in secs, trip distance, pickup longitude, pickup latitude, drop-off longitude, drop-off latitude. To the need of our calculations the string values were encoded as the positions in dictionary, date time values were converted to the numbers of seconds from the beginning of year 2013. We have conducted experiment in two scenarios. In the first one, all the missing values have been filled by zeros. During the latter one, to present the results graphically, the experiments have been conducted for non-empty pickup longitude and latitude geographic positions (737,462 records). The example of data after conversion to numbers is presented in Table 1.

In the first case, for instance, all the records containing the negative values were marked as abnormal (isolated). However, it is difficult to present a concise summary of the results in a graphical form. Hence, we restrict our presentation to the two-dimensional, geographical records. What is important here is that for 99.98% records with zero values replacing the empty fields *k*-Means-based isolation forest returns the minimal possible value, i.e., 0 which means that the records are strictly isolated. On the other hand isolation forest for 99.56% records returns the isolation score about 0.46 which is in-between minimal and maximal values, namely 0.41 and 0.502, respectively. This means that the proposed innovative aspect highly improves the isolation forest giving more intuitive understanding of the results.

In the second case, where only pickup longitude and latitude values were analyzed, the enhancement of IF significantly improves the original version of the method. First, it classifies the outside (in relation to New York City) points as outliers, see Fig. 9. We do not present similar graphics for the whole points area with respect to isolation forest since the points are invisible. Two next plots, Figs. 10 and 11, show that *k*-Means-based version of isolation forest marks the outside points as outliers while isolation forest seeks such kind of points more likely inside the New York area. Here, it is observable that the above-discussed property of *k*-Means-based IF that it looks for outliers more outside the region of interest is important in many areas of applications.

4.3. Intermodal transportation dataset

In the series of experiments presented here, the datasets of train and ship transportation data of one of European intermodal transport companies have been used. There were included two datasets in our experiments. The first was the set of chosen ship transportation data and the second was related to train transports. The first set we have chosen was the data containing 26,384 records with the following attributes: service group code, service type, unit category, unit type, unit kind, ride type id, departure port, arrival port, and the differences between the time the unit has arrived to port and departure, arrival, departure from port, and minimal order og, i.e., minimal realization time. All the categorical string values were converted to the integral values. Similarly, the data containing the following train transportation info have been prepared: service group code, service type, unit category, unit type, unit kind, departure station name, arrival station name, and the differences between the time a unit has arrived the station and train departure, train arrival, unit departure arrival station, and nearest due date. The examples of the data contained in these two datasets are presented in Tables 2 and 3.

Both IF and the proposed approach classifies the records with the values like date train arrival difference, unit departure arrival station difference, and nearest due date difference being negative.

An interesting observation is that the clustering-based method offers much more different rank values (ca. 12,500) of anomaly scores than IF (ca. 1750), and that there are significant differences between these rankings of points in a context of anomaly score, see Figs. 12 and 13. Despite being similar in terms of the classification results, it is easy to note that our proposal is more precise and differentiates between various points. In particular, this fact is noticeable in Table 4. For various records, IF returns the same anomaly scores while our proposal differentiates the final results. Finally, it is worth to note, that both methods treat the categorical data in a similar way. Analogical results were obtained for ship transportation data. Again, the histogram depicted at Fig. 16 shows the differences between the methods' way of work. Interesting insights may be obtained from Figs. 14 and 15, where dependencies between anonymized train routes and the time differences between train departure and arrival are presented. Both IF and *k*-Means-Based IF are highly accurate in searching for anomalies. However, the *k*-Means-based version more likely finds the outliers with limit values.

4.4. Execution times

Here, we discuss computing overhead observed during running both isolation forest and *k*-Means-based isolation forest. All the tests were conducted on the computer equipped with Windows 10 64 bit architecture, 2.4 GB RAM. The application was written in C++. No threads were used to compare the results. The times (see, Table 5) are presented for the sum for 100 iterations of sample tree building and scoring through the whole dataset according to the sample tree with no average.

The tests show an interesting property of *k*-Means-based isolation forest. In most cases, its execution time is quite longer than execution time of the isolation forest. It is because of more comparisons are done at each split and the distances to the cluster centers are yielded. However, in case of large databases containing geographical points it works faster. The cause may be hidden in the structure of data which are relatively clustered at the beginning. Here, it is worth to note that the times of execution are, in fact, negligible from the point of view of data size. The method is called just once for the dataset and, taking into account, the experts' time to analyze the scores, the time of execution is not long, reaching at most less than 18 s in the case of *k*-Means-based isolation forest or less than quarter of an hour in the case of isolation forest — in one case only.

5. Conclusions and future work

In the study, we have presented a new approach to the isolation forest-based methods of outlier detection, which is a novel *k*-Means-based isolation forest method. It is based on the data

Table 2
Example of data from the ship transportation dataset (anonymized).

Service type	Unit category	Unit type	Unit kind	Ride type id	Roro depart. port	Roro arrival port	Roro departure differ.	Roro arrival differ.[a]	Unit departure from port differ.[a]	Min order og differ.	Service type[a]
Feg	Qrs	Uvw	Rpo	Cba	X	Y	33	4854	5687	7188	Feg
Gfe	Rsq	Vwx	Rpo	Dcb	X	Y	274	5095	6409	12 409	Gfe
Fed	Qrs	Uvw	Rpo	Cba	X	Y	201	5022	6336	12 276	Fed
Gfe	Rsq	Vwx	Opr	Cba	X	Z	1576	6397	7711	13 051	Gfe
Fed	Qrs	Wxy	Opr	Dcb	Y	W	859	5680	7234	6574	Fed

[a]Means columns removed from the series of test with 9 attributes.

Table 3
Example of data from the train transportation dataset (anonymized).

Service group code	Service type	Unit category	Unit type	Unit kind	Departure station name	Arrival station name	Date train departure diff.[b]	Date train arrival diff.	Unit departure arrival station diff.[a] [b]	Nearest due date diff.
Abc	Def	Ijl	Kmn	Opr	A	C	470	1245	1640	−3642
Abc	Efg	Jlk	Mno	Opr	B	D	983	2472	3589	1553
Bcd	Def	Jlk	Mno	Prs	A	E	763	1518	7458	3858
Cde	Efg	Lkm	Nop	Prs	B	D	1285	2560	2734	4735
Abc	Fgh	Lkm	Nop	Prs	B	D	983	2472	3247	1553

[a]Mean columns removed from the series of test with 10 attributes.
[b]Mean columns removed from the series of test with 9 attributes.

Table 4
Anomaly score values obtained by the isolation forest and *k*-Means-based isolation forest.

Date train departure dif (min.)	Date train arrival dif (min.)	Unit departure arr station dif (min.)	Nearest due date dif (min.)	*k*-Means-based IF	IF	Rank with *k*-Means IF	Rank with IF
1860	2760	4740	10 860	4.18365	0.448107	795	38
1588	2488	4428	4828	4.18689	0.448107	796	38
1525	2965	3505	8185	4.18711	0.448107	797	38
1566	2466	4406	4806	4.18732	0.448107	798	38
1560	2460	4400	4800	4.18736	0.448107	799	38
1410	2910	9223	8130	4.21224	0.448107	811	38
1479	2379	4319	4719	4.21568	0.448107	812	38
1295	2195	4176	215	4.2277	0.448107	818	38
616	1322	4596	6456	4.24703	0.448107	832	38

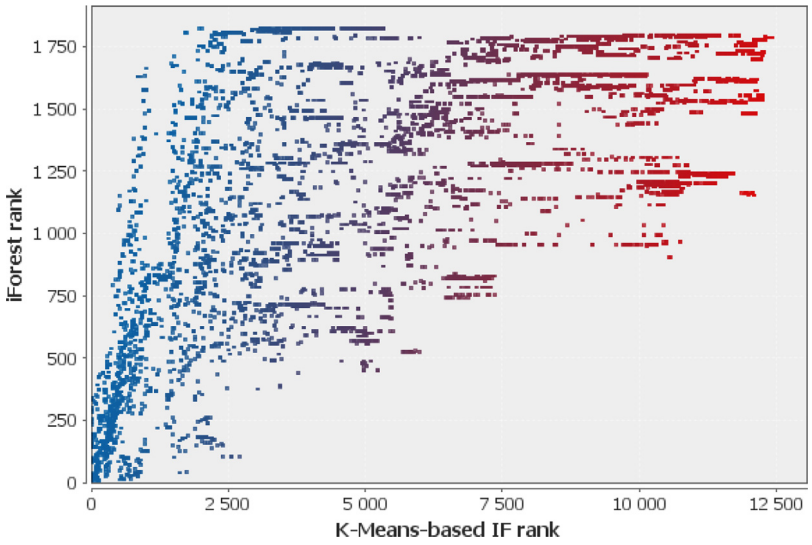

Fig. 13. Modules of rank differences.

structured in a form of trees having at each node an optimal number of leafs. These leafs are created on a basis of analysis of the training dataset filtered at earlier nodes of the tree. During the experiments, the method has demonstrated its efficiency and high level of precision in searching of isolation as well as anomaly records in various kinds of datasets containing geographical points, transportation information, or mixed (including categorical) data.

Future work directions may be an in-depth analysis of the method in other branches of applications, e.g., network anomalies, fraud detection, etc. Moreover, an interesting ways of work seem to be various kinds of Granular Computing related data

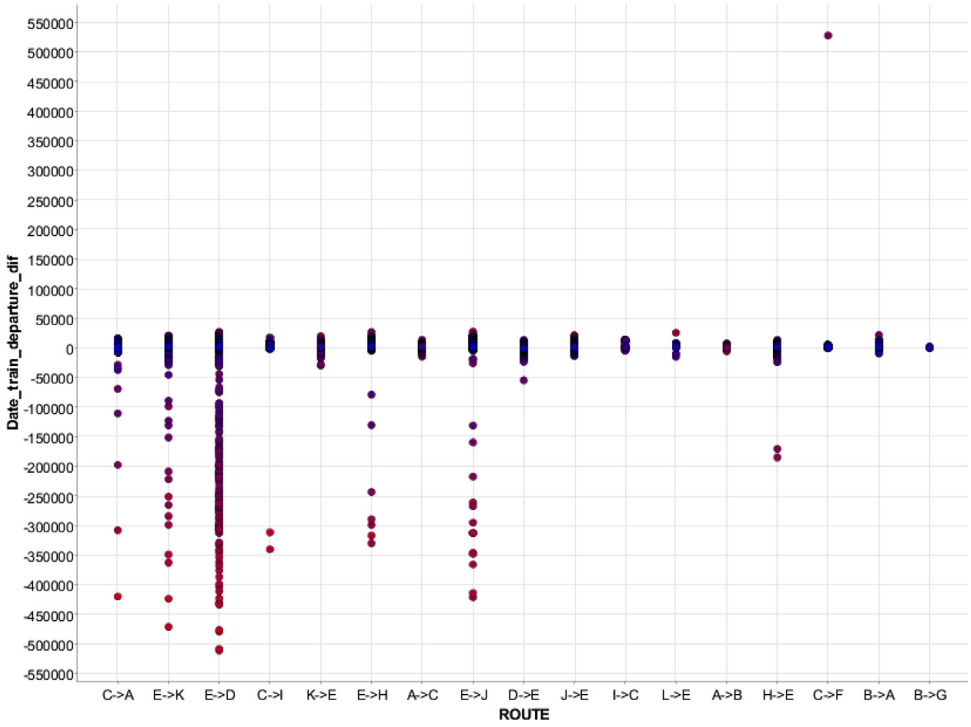

Fig. 14. Relations between anonymized train routes and times between train departures and arrivals to the stations in case of IF. Note that the datasets contains negative values.

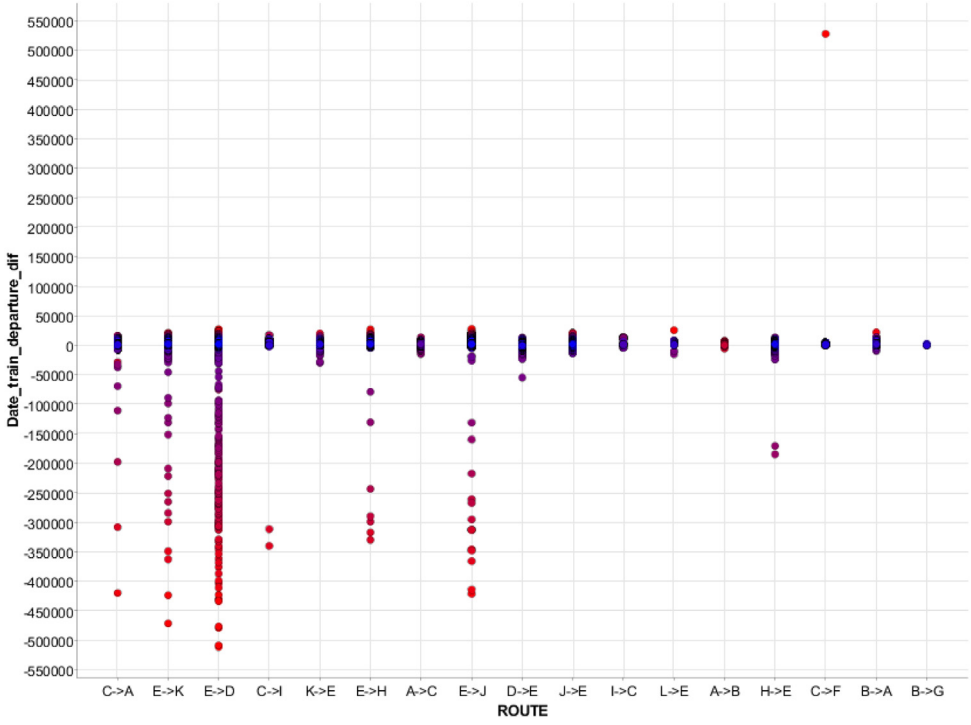

Fig. 15. Relations between anonymized train routes and times between train departures and arrivals to the stations (note that the table contains negative values) in case of *k*-Means-based IF.

structures which may appear in the tree nodes such as fuzzy, rough, or shadowed sets. An application of the method to the task of classification based on an ensemble of classifiers is also worth examining. Finally, an extensive comparative analysis of various clustering-based methods is one of possible directions of further studies.

CRediT authorship contribution statement

Paweł Karczmarek: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Writing - review & editing, Supervision, Project administration, Funding acquisition, Visualization. **Adam Kiersztyn:** Validation, Formal analysis, Investigation, Resources,

Table 5
Computing overhead of the methods.

Dataset kind	Number of records	Number of attributes	Isolation forest (s)	*k*-Means-based isolation forest (s)
Artificial set	30 000	2	4.584	2.524
Artificial set	50 000	2	6.934	2.774
NYC Taxi	3 386 426	14	792.66	17.444
NYC Taxi (geographical positions)	737 462	2	54.687	7.47
Ship transport	79 477	9	14.111	10.691
Ship transport	9890	12	2.142	7.986
Train transport	26 384	11	5.912	7.792
Train transport	26 384	10	5.23	6.792
Train transport	26 384	9	4.723	8.12

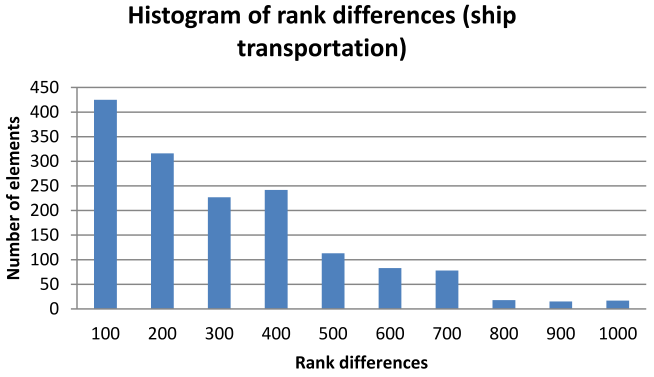

Fig. 16. Rank differences for the data describing ship transportation.

Data curation, Writing - review & editing, Visualization. **Witold Pedrycz:** Conceptualization, Methodology, Validation, Investigation, Formal analysis, Writing - review & editing, Supervision. **Ebru Al:** Resources, Project administration, Funding acquisition, Writing - review & editing.

Acknowledgment

Funded by the National Science Centre, Poland under CHIST-ERA programme (Grant no. 2018/28/Z/ST6/00563).

References

[1] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Comput. Surv. (CSUR) 41 (3) (2009) 1–72.
[2] R.A.A. Habeeb, F. Nasaruddin, A. Gani, I.A.T. Hashem, E. Ahmed, M. Imran, Real-time big data processing for anomaly detection: A survey, Int. J. Inf. Manag. 45 (2019) 289–307.
[3] F.T. Liu, K.M. Ting, Z.-H. Zhou, Isolation forest, in: 2008 Eighth IEEE International Conference on Data Mining, 2008, pp. 413-422.
[4] F.T. Liu, K.M. Ting, Z.-H. Zhou, Isolation-based anomaly detection, ACM Trans. Knowl. Discov. Data (TKDD) 6 (1) (2012) article (3).
[5] F. Angiulli, C. Pizzuti, Fast outlier detection in high dimensional spaces, in: Principles of Data Mining and Knowledge Discovery, in: Lecture Notes in Computer Science, vol. 2431, 2002, pp. 15–26.
[6] H. Kim, Isolation forest step by step, 2019, [online].
[7] S. Ramaswamy, R. Rastogi, K. Shim, Efficient algorithms for mining outliers from large data sets, in: Proceedings of the 2000 ACM SIGMOD Int. Conf. on Management of Data, 2000, pp. 427-438.
[8] E.B. Knorr, R.T. Ng, V. Tucakov, Distance-based outliers: Algorithms and applications, VLDB Int. J. Very Large Data Bases 8 (3–4) (2000) 237–253.
[9] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, R.C. Williamson, Estimating the support of a high-dimensional distribution, Neural Comput. 13 (7) (2001) 1443–1471.
[10] X. Gan, J. Duanmu, J. Wang, W. Cong, Anomaly intrusion detection based on PLS feature extraction and core vector machine, Knowl.-Based Syst. 40 (2013) 1–6.
[11] L. Zhang, J. Lin, R. Karim, Adaptive kernel density-based anomaly detection for nonlinear systems, Knowl.-Based Syst. 139 (2018) 50–63.
[12] E. de la Hoz, E. de la Hoz, A. Ortiz, J. Ortega, A. Martínez-Álvarez, Feature selection by multi-objective optimisation: Application to network anomaly detection by hierarchical self-organising maps, Knowl.-Based Syst. 71 (2014) 322–338.
[13] S.M. Erfani, S. Rajasegarar, S. Karunasekera, C. Leckie, High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning, Pattern Recognit. 58 (2016) 121–134.
[14] P. Malhotra, L. Vig, G.G. Shroff, P. Agarwal, Long short term memory networks for anomaly detection in time series, in: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2015, pp. 89-94.
[15] C. Zhou, R.C. Paffenroth, Anomaly detection with robust deep autoencoders, in: KDD '17 Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, 2017, pp. 665-674.
[16] R.J.G.B. Campello, D. Moulavi, A. Zimek, J. Sander, Hierarchical density estimates for data clustering, visualization, and outlier detection, ACM Trans. Knowl. Discov. Data 10 (1) (2015) 5.
[17] Z. He, X. Xu, S. Deng, Discovering cluster-based local outliers, Pattern Recognit. Lett. 24 (9–10) (2003) 1641–1650.
[18] R. Scitovski, K. Sabo, DBSCAN-like clustering method for various data densities, Pattern Anal. Appl. (2019) http://dx.doi.org/10.1007/s10044-019-00809-z.
[19] Z. Wu, J. Huang, Application of DBSCAN cluster algorithm in anormaly detection, Netw. Comput. Secur. 8 (2007) 43–46.
[20] J. Li, X. Hu, Efficient mixed clustering algorithm and its application in anomaly detection, J. Comput. Appl. 7 (2010) 1916–1918.
[21] W. Chimphlee, A.H. Abdullah, M.N.M. Sap, S. Srinoy, S. Chimphlee, Anomaly-based intrusion detection using fuzzy rough clustering, in: 2006 International Conference on Hybrid Information Technology, Cheju Island, 2006, pp. 329-334.
[22] J. Gomez, F. Gonzalez, D. Dasgupta, An immuno-fuzzy approach to anomaly detection, in: The 12th IEEE International Conference on Fuzzy Systems, FUZZ '03., Vol. 2. St Louis, 2003, pp. 1219-1224.
[23] X.D. Hoang, J. Hu, P. Bertok, A program-based anomaly intrusion detection scheme using multiple detection engines and fuzzy inference, J. Netw. Comput. Appl. 32 (6) (2009) 1219–1228.
[24] C.-H. Tsang, S. Kwong, H. Wang, Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection, Pattern Recognit. 40 (2007) 2373–2391.
[25] J. Liu, J. Tian, Z. Cai, Y. Zhou, R. Luo, R. Wang, A hybrid semi-supervised approach for financial fraud detection, in: 2017 International Conference on Machine Learning and Cybernetics (ICMLC), Ningbo, 2017, pp. 217-222.
[26] H. Izakian, W. Pedrycz, Anomaly detection in time series data using a fuzzy c-means clustering, in: 2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), Edmonton, AB, 2013, pp. 1513-1518.
[27] H. Izakian, W. Pedrycz, I. Jamal, Clustering spatiotemporal data: An augmented fuzzy c-means, IEEE Trans. Fuzzy Syst. 21 (5) (2013) 855–868.
[28] H. Izakian, W. Pedrycz, Anomaly detection and characterization in spatial time series data: A cluster-centric approach, IEEE Trans. Fuzzy Syst. 22 (6) (2014) 1612–1624.
[29] U. Yun, H. Ryang, G. Lee, H. Fujita, An efficient algorithm for mining high utility patterns from incremental databases with one database scan, Knowl.-Based Syst. 124 (2017) 188–206.
[30] H. Faris, A.M. Al-Zoubi, A.A. Heidari, I. Aljarah, M. Mafarja, M.A. Hassonah, H. Fujita, An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks, Inf. Fusion 48 (2019) 67–83.
[31] P. Protopapas, J.M. Giammarco, L. Faccioli, M.F. Struble, R. Dave, C. Alcock, Finding outlier light curves in catalogues of periodic variable stars, Mon. Not. R. Astron. Soc. 369 (2) (2006) 677–696.
[32] E. Keogh, J. Lin, A. Fu, Hot SAX: Efficiently finding the most unusual time series subsequence, in: Proc. Fifth IEEE Int. Conf. Data Mining, 2005, pp. 226-233.
[33] E. Keogh, S. Lonardi, C.A. Ratanamahatana, Towards parameter-free data mining, in: Proc. the 10th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, 2004, pp. 206-215.
[34] M. Das, S. Parthasarathy, Anomaly detection and spatio-temporal analysis of global climate system, in: Proc. Third Int. Workshop on Knowledge Discovery from Sensor Data, 2009, pp. 142-150.
[35] D. Gao, Y. Kinouchi, K. Ito, X. Zhao, Neural networks for event extraction from time series: A back propagation algorithm approach, Future Gener. Comput. Syst. 21 (7) (2005) 1096–1105.
[36] D. Dasgupta, S. Forrest, Novelty detection in time series data using ideas from immunology, in: 5th Int. Conf. on Intelligent Syst., 1996.
[37] J. Ma, S. Perkins, Time-series novelty detection using one-class support vector machines, in: Proc. Int. Joint Conf. Neural Networks, 2003, pp. 1741-1745.

[38] M. Du, F. Li, G. Zheng, V. Srikumar, DeepLog: Anomaly detection and diagnosis from system logs through deep learning, in: CCS '17 Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 1285-1298.
[39] S. Kanarachos, S.-R.G. Christopoulos, A. Chroneos, M.E. Fitzpatrick, Detecting anomalies in time series data via a deep learning algorithm combining wavelets, neural networks and Hilbert transform, Expert Syst. Appl. 85 (1) (2017) 292–304.
[40] M. Munir, S.A. Siddiqui, A. Dengel, S. Ahmed, Deepant: A deep learning approach for unsupervised anomaly detection in time series, IEEE Access 7 (2019) 1991–2005.
[41] G. Box, G.M. Jenkins, G.C. Reinsel, G.M. Ljung, Time Series Analysis: Forecasting and Control, Wiley, 2015.
[42] J. Takeuchi, K. Yamanishi, A unifying framework for detecting outliers and change points from time series, IEEE Trans. Knowl. Data Eng. 18 (4) (2006) 482–489.
[43] H. Cheng, P. Tan, C. Potter, S. Klooster, A robust graph-based algorithm for detection and characterization of anomalies in noisy multivariate time series, in: Proc. IEEE Int. Conf. Data Mining Workshops, Pisa, Italy, 2008, pp. 349-358.
[44] A. Khatkhate, A. Ray, E. Keller, S. Gupta, S.C. Chin, Symbolic time series analysis for anomaly detection in mechanical systems, IEEE Trans. Mechatron. 11 (4) (2006) 439–447.
[45] E.W. Dereszynski, T.G. Dietterich, Spatio-temporal models for data anomaly detection in dynamic environmental monitoring campaigns, ACM Trans. Sens. Netw. 8 (1) (2011) 3.
[46] D.J. Hill, B.S. Minsker, E. Amir, Real-time Bayesian anomaly detection for environmental sensor data, in: Proc. 32nd Congress of the Int. Assoc. of Hydraulic Eng. and Research, 2007.
[47] D.B. Neill, Expectation-based scan statistics for monitoring spatial time series data, Int. J. Forecast. 25 (3) (2009) 498–517.
[48] J.C. Bezdek, R. Ehrlich, W. Full, FCM: The fuzzy c-means clustering algorithm, Comput. Geosci. 10 (2–3) (1984) 191–203.
[49] V. Chandola, V. Mithal, V. Kumar, Comparative evaluation of anomaly detection techniques for sequence data, in: 8th IEEE Int. Conf. on Data Mining, Pisa, Italy, 2008, pp. 743-748.
[50] P. D'Urso, R. Massari, Fuzzy clustering of mixed data, Inform. Sci. 505 (2019) 513–534.
[51] L. Akoglu, H. Tong, D. Koutra, Graph based anomaly detection and description: a survey, Data Min. Knowl. Discov. 29 (3) (2015) 626–688.
[52] T. H. Fanaee, J. Gama, Tensor-based anomaly detection: An interdisciplinary survey, Knowl-Based Syst. 98 (2016) 130–147.
[53] R.L. Thorndike, Who belongs in the family?, Psychom. 18 (4) (1953) 267–276.
[54] D.J. Ketchen Jr, C.L. Shook, The application of cluster analysis in strategic management research: An analysis and critique, Strateg. Manag. J. 17 (6) (1996) 441–458.
[55] J.A. Hartigan, M.A. Wong, A K-Means clustering algorithm, Appl. Stat. 28 (1) (1979) 100–108.
[56] B. Donovan, D. Work, New York City Taxi Trip Data (2010-2013), University of Illinois at Urbana-Champaign, 2016.