TELEINFORMATICS REVIEW
No. 1-4, 2023

Unsupervised machine learning in financial anomaly detection: clustering algorithms vs. dedicated methods

Radosław J. WOŹNIAK

Institute of Information Systems, Faculty of Cybernetics, MUT
Kaliskiego 2, 00-908 Warsaw, Poland
radoslaw.wozniak@wat.edu.pl

ABSTRACT: The article presents the application of selected clustering algorithms for detecting anomalies in financial data compared to several dedicated algorithms for this problem. To apply clustering algorithms for anomaly detection, the Determine Abnormal Clusters Algorithm (DACA) was developed and implemented. This parameterized script (DACA) allows clusters containing anomalies to be automatically detected on the basis of defined distance measures. This kind of operation allows clustering algorithms to be quickly and efficiently adapted to anomaly detection. The prepared test environment has allowed for the comparison of selected clustering algorithms. K-Means, Hierarchical Cluster Analysis, K-Medoids, and anomaly detection: Stochastic Outlier Selection, Isolation Forest, Elliptic Envelope. The research has been carried out on real financial data, in particular on the income declared in the asset declarations of the targeted professional group. The experience of financial experts has been used to assess anomalies. Furthermore, the results have been evaluated according to a number of popular classification and clustering measures. The highest result for the investigated financial problem was provided by the K-Medoids algorithm in combination with the DACA script. It is worthwhile to conduct future research on the introduced solutions as an ensemble method.

KEYWORDS: Anomaly detection, Classification, Clustering, Financial fraud, Finance

1. Introduction

The problem of fraudulent activities, irregularities, or financial fraud is very complicated and occurs in many areas of daily life. In the financial aspect, fraud may involve the manipulation of various documents, falsification of assets, violation of accounting policies, operating fake transactions, or concealing them by omitting important records from the books. Familiarity with the problem, the

rules, the characteristic features, and the common parts could make it possible to take measures to detect, prevent, and counter fraud.

One very advanced area that stands out in the fight against financial crime is the banking sector. Highlights, among other things, two types of systemic solutions to counter crime. The AML (Anti Money Laundering) system to counter money laundering and terrorist financing, and the FDS (Fraud Detection System) to detect fraud, malpractice, and financial fraud by identifying suspicious transactions, transfers, and other fraudulent activities. These systems, among others, analyse and assess transaction risks, identify suspicious customers, monitor them, and process and report information required by governments. The other features are characterised by solutions in the tax area. These include tools to counter exploitation and mitigate opportunities for tax fraud [6], [15], [18].

The article addresses unsupervised machine learning algorithms that, individually or as an ensemble method, have the potential to support processes involving the detection of financial and asset irregularities or fraud [3], [20]. The scope of information that is analysed includes: property, financial resources, mobile properties, financial obligations, revenues, salaries, expenses, membership in several types of companies and organisations, business management, personal and professional information of themselves and their spouse. The aforementioned range of information can be described as assets, which are included in asset declarations. In various countries, these documents are submitted by personnel who are obligated to submit them by laws relating mainly to political life or selected uniformed services [4]. The research was elaborated on the basis of unpublished anonymised data from an organisation authorised to process the aforementioned information.

2. Unsupervised machine learning algorithms implemented

Unsupervised learning operates on data without specific labels, classes, or known output values. It is used for grouping data with similar characteristics, finding patterns, outliers in a set based on specific features, similarities, and distances in the input dataset. Due to the highly specialised nature of the studied data and the lack of specific labels, i.e. classes that evaluate the financial values of large data sets, the following section describes and applies a selected assortment of unsupervised learning algorithms for clustering and anomaly detection. More than 20 clustering and anomaly detection algorithms have been analysed in a broader study, from the range of distance-based, neighbourhood-based, probabilistic, statistical, neural network-based, domain-based, isolation-based and other methods. In this paper, distance-based methods (Hierarchical Cluster Analysis, K-Means and K-Medoids) and representatives of statistical methods (Stochastic Outlier Selection), isolation-based (Isolation Forest) and other

methods (Elliptic Envelope) are presented and used. These algorithms represent different and distinctive approaches to anomaly detection and present some of the best average anomaly detection results (e.g., accuracy) on the financial data examined. Furthermore, many of them present the same expected observations as anomalies. But interesting cases for further analysis by financial experts are observations that are on the borderline between normal and anomalous, so that they are labelled as anomalies by one algorithm and not by another.

The clustering algorithms (Hierarchical Cluster Analysis, K-Means, and K-Medoids) were adapted to the anomaly detection problems explored using the Determine Abnormal Clusters Algorithm (DACA). DACA and other known approaches to using clustering for anomaly detection, e.g., CBLOF (Cluster-Based Local Outlier Factor) [7] and LDCOF (Local Density Cluster-based Outlier Factor) [24] have been discussed in the next section.

The other anomaly detection algorithms (Stochastic Outlier Selection, Isolation Forest, and Elliptic Envelope) were used as self-independent methods, without using the aforementioned DACA.

For additional verification of the anomaly detection results of the presented algorithms, one of the simple baseline algorithms, such as k-Nearest Neighbors, was used to evaluate the level of anomalies in the whole set.

All the algorithms mentioned and implemented are briefly described in the following.

2.1. K-Means (K-M)

The K-Means, known as K-Centroids. It aims to divide an n-element set of observations (also known as elements, points, samples, values, among others) into k subsets of observations (so-called clusters, also known as clusters, groups), in which each observation is assigned to the nearest cluster, according to the criterion of similarity - distance from the mean value, representing the geometric centre of the cluster, or so-called centroid. It is an example of partitioned clustering, which means that the entire input data set is divided into groups. Clustering is an optimisation problem that involves minimising the within-group variance or sum of squared errors (SSE). The K-Means algorithm requires a predetermined number of clusters, and it tends to build clusters of similar size. The partitioning of space into clusters is related to Voronoi's theory of commons [17], [18].

All of the presented clustering algorithms (Hierarchical Cluster Analysis, K-Means and K-Medoids) have wide application in many disciplines, and not only in the field of anomaly detection [1].

2.2. Hierarchical Cluster Analysis (HCA)

The Hierarchical Cluster Analysis (HCA); otherwise, Hierarchical Clustering builds a hierarchical, tree-like structure of clusters, which can be visualised by a dendrogram (binary tree). The root of such a tree defines a cluster that contains all objects, while its leaves represent clusters with one observation. On the other heights of the tree are clusters, grouping observations, which are also other clusters. The algorithm does not require specifying the number of clusters to perform internal calculations and compute the dendrogram. However, to qualify the data into a specific number of clusters, it is necessary to specify the number of clusters, which is reflected in the height of the dendrogram at which the tree will be pruned [10], [25].

To build clusters, various distance metrics are utilised to calculate the level of dissimilarity between observations. Two hierarchical cluster building strategies can be distinguished depending on the direction of cluster building: agglomerative clustering (hierarchical bottom-up, bottom-up strategy), and deglomerative clustering (hierarchical top-down, top-down strategy). Moreover, in the case of agglomerative clustering, several subtypes of algorithms can be distinguished on the basis of the type of linkage. In this paper, Ward's linkage method is implemented, where pairs of clusters are combined based on the criterion of minimising the sum of squares of errors within the clusters [12].

2.3. K-Medoids (KMed)

The K-Medoids is an example of partitioned clustering, an algorithm similar in principle of operation to K-Means. The algorithm splits the input data set containing observations into subsets, so-called clusters (groups). It assigns each observation to one of the clusters according to the criterion of distance to the element known closest to the centre of the group, here the so-called medoid. In this case, the value of the absolute error criterion, called the total deviation, is optimised. The algorithm requires a predefined number of clusters [12].

Compared to K-Means, the K-Medoids algorithm is relatively more resilient to outliers (including noise). Medoids are used (instead of centroids), real observations as a balanced centre of clusters. This increases the level of stability of the clusters, the sets are less susceptible to change, and it allows to indicate the representative points of each cluster. In contrast, K-Means is faster and more efficient for large data sets. [21].

2.4. Stochastic Outlier Selection (SOS)

The Stochastic Outlier Selection is an algorithm that applies the concept of affinity to quantify the relationship between the observations examined. The quoted affinity function is proportional to the similarity between neighbouring observations. The lower the affinity, the greater the diversity between observations. The anomalies are typed by the low affinity of an observation relative to the other elements (observations). The algorithm may closely resemble nearest-neighbour methods in its behaviour [8].

Stochastic outlier selection as well as the next Isolation Forest algorithm have significant applications for anomaly detection in many scientific fields (eg medicine [5]).

2.5. Isolation Forest (IF)

The Isolation Forest is an anomaly detection algorithm that has a different approach to popular methods. The algorithm first identifies and isolates anomalies from normal observations, instead of traditionally identifying a set of normal observations at the start . The method is an unsupervised version of a decision tree and utilises a binary decision tree for its operation. The algorithm performs cyclic random divisions of the data space into two parts. The observations suitable for anomalies and the values of the separation between the minimum and maximum values of these observations are randomly selected. The algorithm is based on the assumption that anomalies are less frequent in the set and that anomalies belong to shallow branches, so that they are noticeably isolated from the rest of the tree branches. An observation that is high (deep) in the tree is unlikely to be an outlier. The algorithm can additionally use a sub-sampling process in its operation, randomly selecting smaller samples of data for analysis. Such action allows for a more accurate detection of anomalies in large data sets [16].

According to the authors, the solution is computationally efficient, achieves good results for multidimensional datasets, and has a high efficiency compared to LOF or random forest algorithms. The final performance depends on the adopted contamination parameter [21].

Isolation Forest have significant applications for anomaly detection in many scientific and business fields (e.g., finance [9], medicine [5]).

2.6. Elliptic Envelope (EE)

The Elliptic Envelope is an algorithm for anomaly detection using the titled elliptic envelope. The key assumption is that the input data set has an approximate distribution to the Gaussian distribution. Observations classified as normal values

occur in the high-probability region of this distribution, while anomalies with the opposite distribution or significantly do not fit this distribution. This method outlines an ellipse based on an estimate of the covariance of observations relative to the localised principal central elements (observations). To mark the boundary of the ellipse, a distance measure can be used on the basis of a predefined hyper-parameter indicating the percentage number of anomalies. The problem in this case is to fine-tune this parameter [22].

Elliptic Envelope have significant applications for anomaly detection in many business fields (eg finance [23]).

2.7. k-Nearest Neighbors (kNN)

The k-Nearest Neighbors (k-NN) algorithm is a supervised machine learning method that is used for classification and regression. The principle of operation is based on similarity, similar distance of points from each other. For a new point, the same label is assigned as for other points nearby. Therefore, for a new observation, the distance to all points in the whole set is calculated. Next, the k nearest neighbors are selected. Based on these, a classification (e.g., by majority voting) of the new point is carried out, that is, a label is given to the new observation.

The k-Nearest Neighbors algorithm has been adapted for anomaly detection by adding several improvements, including a contamination parameter that determines the expected number of anomalies. The purpose of k-Nearest Neighbors Detector is unsupervised identification of points that are significantly different from the rest of the data. The evaluation of anomalies is based on the distances to k nearest neighbors. Three detectors are supported as anomaly measures: largest - the largest distance, i.e. the distance to the farthest neighboring k observations, is used; mean - the average distance of the neighboring observations is used; median - similarly, the median is used. The solution uses an internal algorithm to calculate nearest neighbors, to expand the so-called tree is BallTree [1].

3. Determine Abnormal Clusters Algorithm (DACA)

Unsupervised learning algorithms for cluster building have the potential to be applied to the detection of anomalies, a set of observations that are abnormal, non-standard, mismatched or deviate under certain criteria from the rest of the elements. The developed authorship script DACA (Determine Abnormal Clusters Algorithm) allows the application of clustering or other algorithms that divide

observations into groups for purposes such as anomaly detection. The algorithm works in cooperation with clustering algorithms and requires a previously performed partitioning of a set of observations into clusters.

In such cases, you may be able to encounter other methods that detect anomalies in grouped sets like CBLOF (cluster-based local outlier factor) [7] and LDCOF (local density cluster-based outlier factor) [24], but they function locally, searching for anomalies within each cluster. The first, CBLOF, operates based on distance within a cluster (the product of the distance of observations from the group centroid and the size of the group in question), while the second, LDCOF, operates based on the density of each observation in the cluster.

The DACA algorithm acts globally, at the level of clusters, rather than at the level of individual points. DACA allows automatic determination of clusters that are expected to contain the most anomalies. The functioning principle is to select clusters as whole sets of anomalies based on the distance of the cluster centroid from the balance point of the whole set. The greater the distance, the higher the certainty that a certain cluster includes more anomalies. The clusters (cluster centroids) are sorted according to the distance criterion from the balance points of the whole set, which makes it possible to label successive clusters as anomalous until a designated threshold for the number of anomalies (the sum of the sizes of the labelled clusters) is reached. The algorithm requires the determination of a threshold percentage of the minimum content of the number of anomalies (parameter ap - anomaly percentage), analogous to the "contamination" parameter in many anomaly detection algorithms (e.g., Stochastic Outlier Selection, Isolation Forest, Elliptic Envelope). It is worth noting that the complexity of the main steps of the algorithm is linear O(n), where n is the number of clusters, but the quicksort algorithm used increases the optimistic complexity of the algorithm to the log-linear level O(log n). The algorithm provides a fast and efficient way to adapt clustering algorithms to find and label clusters as normal and abnormal, that is, to detect anomalies.

The algorithm reaches the best results for a properly prepared dataset, e.g. filtered by appropriate criteria. This approach allows for searching for anomalies that depend on defined contexts. Note that the described global operation of the algorithm at the cluster level has its pros and cons. The algorithm makes it possible to achieve high sensitivity in most cases but sometimes with a decrease in accuracy for certain situations, e.g. with inadequate filtering of input data. The algorithm is sensitive to the value of the parameter of the percentage number of anomalies. Incorrect adjustment of the value of this parameter can consequently lead to marking as anomalous the whole cluster which contains no anomalies or a small number of anomalies. Adjust the value of this parameter depending on the character of the data. In tested situations with contextual filtering for financial data, the DACA algorithm, in cooperation with many clustering algorithms (e.g., K-Medoids), reaches better results of classification quality measures than anomaly detection algorithms.

The general steps of the DACA algorithm are the following:

1. Verify and store the current number of clusters in the variable "k" (number of clusters). There must be at least 2 clusters, the number of clusters depends on the size and character of the data, the default recommended number of groups is a minimum of 8. Determination of the balance point of the whole set; it is a mathematically calculated balance point for a collection, or it may be a specific value due to the character of the data, e.g., in the case of specific financial data, it might be 0 (e.g., for two dimensions it will be 0.0, etc.).).
2. Create a new temporary "cluster" array. The columns of the array are: cluster number, cluster centroid ("centre point of the cluster"), distance of the cluster centroid from balance point of the whole set, cluster count, new cluster number, classification of the cluster in the context of anomalies (default value 0 means normal cluster, 1 is anomalous). Initialisation of the auxiliary iteration variable "i" with a value of zero to move through subsequent clusters. Initialisation of the "sum" variable, the total sum of anomalies with a value of zero.
3. Compute the following values for each cluster (according to the order of the cluster number) with all its elements: the cluster centroid (analogous to the K-Means algorithm), the distance of the cluster centroid from the balance point of the whole set, the cluster count. These values are then stored as subsequent rows about the clusters in a temporary array "Cluster", in equivalent columns.
4. Sort the "Cluster" array in ascending order according to the column concerning the distance of the cluster centroid from the balance point of the whole set and the overall location of the cluster. The implementation uses the popular quicksort algorithm and the Euclidean metric.
5. Assign a new cluster number (counting upwards from zero inclusive) to each cluster in the "Cluster" array according to the new sorted order. Following the new numbering, a cluster with an increasing number represents observations located farther and farther away from the balance point of the whole set.
6. Mark the clusters as normal and abnormal according to the specified percentage indicator (default 15%) of the minimum number of anomalies sought. Until the percentage indicator is reached, subsequent clusters starting from the largest new cluster number are marked as anomalous - a value of 1 is assigned in the column regarding the evaluation of the cluster in the context of anomalous. As you proceed past the next farthest clusters, the variable "sum" of anomalies, or the total number of currently marked anomalies, increases. The percentage indicator serves as a condition for the end of anomaly labelling. After the number of clusters larger than required is counted, the loop stops. It is useful to add that the percentage of the minimum number of anomalies searched is the own

parameter "ap" (from anomaly percentage). This is a key parameter for the clustering algorithms presented applied for anomaly detection.

7. The algorithm is finished, the results are prepared, and the variables are cleaned.

4. Methodology

4.1. Background

For the purpose of the article, a set of research data was provided. The scope of information listed in the Introduction is very wide, and the explanation of individual assets would require a minimum of several separate articles. Asset items such as income and other revenue of a targeted group of people were used as input data. This means that the investigation was conducted for a limited and selected dependent group of a specific context [4]. The input data was cleaned and prepared for analysis. The process involved the application of a number of operations, including: combining data from different sources, handling missing data, normalisation, generating new features (feature engineering) and also preparing data according to business (fintech) rules, e.g. in relation to tax law on income, expenses, profits. A group of experts in the field of financial data evaluated the data for anomalies; such an effort will allow the results to be evaluated using classification measures.

Finally, each algorithm was probed on identical data sets to compare the results. It is worth highlighting that the research data set has been prepared based on actual data.

4.2. Parameters of the algorithms

Python implementations of the scikit-learn library version 15.0.2 (released on July 3, 2024 - currently the latest version), the scikit-learn-extra library version 0.3.0 (released on March 27, 2023 - currently the latest version), the pyod library version 2.0.1 (released on August 15, 2024 - currently almost the latest version), were deployed. Each implantation has its own set of parameters. Generally speaking, the setting possibilities are numerous and the algorithms have been tested in a number of possible ways. The article discusses only a few parameters selected for each algorithm. And the results table shows only 3 selected parameter variations for each algorithm.

For Hierarchical Cluster Analysis (HCA), the AgglomerativeClustering class from the sklearn.cluster module of the scikit-learn has been implemented, variants of the algorithm are presented depending on the parameters: "n" and "ap". The settings of the other parameters are: "m"(metric) = "Euclidean"; "l"(link) = "Ward". Description of the parameters used:

1. Parameter "n" (from n_clusters), the number of intended clusters.
2. Additional own DACA parameter "ap" (from anomaly percentage), the percentage of the minimum number of anomaly searches used to select and aggregate clusters that comply with the conditions for identification as anomalous.
3. Parameter "m" (from metric, formerly affinity), a variable that determines the distance measure used for the calculation.
4. Parameter "l" (from linkage), a variable specifying the linkage used. The default value is "ward." Other values are: "complete", "average", and "single".

For K-Means (K-M), the KMeans class from the sklearn.cluster module of the scikit-learn library has been implemented; variants of the algorithm are presented depending on the parameters: "n" and "ap", and the Euclidean metric was used for the calculation. The settings of the other parameters are: "a"(method) = "Lloyd", "i"(init) = "k-means++". Description of the parameters used:

1. Parameter "n" (from n_clusters), as above with HCA.
2. DACA parameter "ap" " as above in HCA.
3. Parameter "a" (from algorithm), a variable specifying the core internal algorithm used. The default value is "Lloyd".
4. Parameter "i" (from init), a variable specifying how the centroid is initialised for all clusters. The default value is "k-means++" (the algorithm uses the empirical probability distribution of observations in the motionless inertia phase).

For K-Medoids (KMed), the KMedoids class from the sklearn_extra.cluster module of the scikit-learn-extra library has been implemented, variants of the algorithm are presented depending on the parameters: "n", "ap" and parameter pairs "i"(init)="k-medoids++" and "a"(method)="pam", "i"(init)="random" and "a"(method)="alt"(alternate). Parameter description similar to that for K-Menas.

For Stochastic Outlier Selection (SOS), the SOS class from the pyod.models module of the pyod library has been implemented, and variants of the algorithm depending on the parameters "c" and "p" are presented. Description of the parameters used:

1. Parameter "c" (from contamination), the percentage of contamination, that is, the coefficient of anomalies in a set of observations. The coefficient defines the maximum number of anomalies. The default value of c=0.1. The range of values (0; 0.5].

2. Parameter "p" (from perplexity), translated as perplexity, is a coefficient that defines the effective number of adjacent observations treated as normal. For the purposes of the exercise, presented as a percentage.

For Isolation Forest (IF), the IsolationForest class from the sklearn.ensemble module of the scikit-learn library has been implemented, variants of the algorithm are presented depending on the parameters "ms": and "c". Description of the parameters used:

1. Parameter "ms" (from max samples), the number of observations taken to learn each baseline estimator. Default value "auto" (number of observations or max 256).
2. Parameter "c" (from contamination), as above with SOS. The default value is "auto." The remaining range of values (0; 0.5].

For Elliptic Envelope (EE), the EllipticEnvelope class from the sklearn.covariance module of the scikit-learn library has been implemented, variants of the algorithm depending on the parameter "c" are shown, where the parameter "c" (from contamination) is the percentage of contamination, as above with IF and SOS. The default value is 0.1. The range of values is (0; 0.5].

For k-Nearest Neighbours (kNN), the kNN class from the models.knn module of the pyod library has been implemented, variants of the algorithm depending on the parameters "c" and "n" are presented, the Euclidean metric was used for the calculation. The remaining parameters have default values. Description of the parameters used:

1. Parameter "c" (from contamination), as above with IF, SOS, and EE. The default value is 0.1.
2. Parameter "n" (from n_neighbors), number of neighbors to utilise by the queries of k neighbors queries. The default value is 5.

4.3. Classification and clustering quality measures

In order to compare and evaluate the effectiveness of the different algorithms, quality classification and clustering measures were utilised [11], [14], such as:

1. (%Anom) - Percentage of anomalies, the percentage of all observations labelled as anomalies.
2. (Sen) - Sensitivity, recall, the percentage of true labelled anomalies relative to total anomalies.
3. (FPR) - Fall-out, FPR - the percentage of labelled false anomalies relative to all normal ones. This assessment has additional important implications. Labelled anomalies should be proposed for further analysis.
4. (F1) - F1 score - the percentage of the harmonic mean of precision and sensitivity.
5. (Acc) Accuracy - the percentage of correctly labelled anomalies and as normal.

6. (AveSil) Average Silhouette - the average silhouette width for the entire dataset is calculated as the average silhouette width of all observations. This is a very general assessment of clustering quality, the higher the value, the better the clustering quality.
7. (%pSil) Percentage of Positive Silhouette - the percentage of positive silhouette values from the set of all observation silhouettes. The higher value means that most observations are well-fit to their clusters.

The above classification and clustering quality ratings are intended to characterise the entirety of the binary classified results, as anomalies or normal observations.

In addition to the article, in further analysis of the results, when the strength of individual anomalies is assessed, the value of the anomaly score (decision function) for each observation is taken into account. For the anomaly detection methods used, the built-in anomaly score (the "score" available in the library) can be used. For clustering algorithms using DACA, in the basic version (described in the article), there is only a label that indicates that the evaluated observation is normal or anomalous. In the next expanded version of DACA, an anomaly score would have to be added. Each observation could be scored by the multiplication of the normalised distance from the whole set's balance point and the cluster weight. In this case, the cluster weight is a value that depends on the sorted clusters from normal (no anomalies, the value of the weight goes to 0) to anomalous (containing anomalies, the value of the weight goes to 1).

The anomaly score represents the degree to which an instance is considered abnormal. This makes it possible to sort the observations according to the level of anomalies (anomaly score, decision function), thus indicating the possible order in which the anomalies might be analysed by specialists. This allows the specialist to select the number of top observations with the strongest anomaly scores as anomalies, or to apply an anomaly strength threshold to identify the most significant anomalies.

5. Experiment and Results

Based on the algorithms introduced and a number of variant parameters, research was carried out to compare their performance in terms of anomaly detection. The analyses were performed on actual anonymised financial data from a targeted professional group. For the purpose of this article, only a fragment of a large dataset with many features described earlier in the Introduction was used. The study used 1000 observations, all data are numerical values, two dimensions on the processed final values (in terms of calculated revenues, expenses, profits from specific time periods), and the number of anomalies is 20%. The data size

used is strictly limited by the need for manual evaluation by financial professionals.

All algorithms tested an identical data set. To evaluate the classification measures, the chosen set of observations was manually assessed by experts from the finance department, who judged the benchmark anomalies as attention-worthy or potentially suspicious (for clarification, the algorithms were trained on the subset without human ratings and evaluated on the subset with human ratings).

Six unsupervised machine learning algorithms were selected for the research, clustering algorithms in association with the DACA (Hierarchical Cluster Analysis, K-Means and K-Medoids, and anomaly detection methods (Stochastic Outlier Selection, Isolation Forest, Elliptic Envelope). All of them have positive results and opinions in the context of anomaly detection and have reached a high sensitivity and accuracy in terms of the various financial data studied (mentioned in Section 2).

Furthermore, the appropriate parameter values were selected to enhance the performance of anomaly detection. A list of variant algorithms with different parameter values is presented below (numerical and letter numbering as shown in Tab. 1, where the letters A-C represent subsequent "Var." variants of a given algorithm):

1. A. HCA, n=10, l=ward, m=eucl, ap=0.15.
2. B. HCA, n=15, l=ward, m=eucl, ap=0.15.
3. C. HCA, n=15, l=ward, m=eucl, ap=0.3.
4. A. K-Means, n=15, i=K-Means++, a=full, ap=0.15.
5. B. K-Means, n=18, i=K-Means++, a=full, ap=0.2.
6. C. K-Means, n=25, i=K-Means++, a=full, ap=0.15.
7. A. K-Medoids, n=15, i=k-medoids++, a=pam, ap=0.2.
8. B. K-Medoids, n=10, i=random, a=alt, ap=0.15.
9. C. K-Medoids, n=25, i=random, a=alt, ap=0.2.
10. A. SOS, c=0.2, pp=0.2.
11. B. SOS, c=0.2, pp=0.25.
12. C. SOS, c=0.2, pp=0.3.
13. A. IF, n=100, ms=auto, c=auto.
14. B. IF, n=100, ms=auto, c=0.3.
15. C. IF, n=100, ms=0.1, c=0.2.
16. A. EE, c=0.1.
17. B. EE, c=0.15.
18. C. EE, c=0.2.
19. A. kNN, n=10, c=0.15.
20. B. kNN, n=10, c=0.2
21. C. kNN, n=10, c=0.25.

The results of the classification and clustering quality measures for the following algorithm variants are presented in the table below (in the headings of

the columns, abbreviated names of measures are provided; also, the order of measures is as described in the earlier section).

Tab. 1. Results of Quality Measures

Nr	Var.	Alg.	%Anom	Sen	FPR	F1	Acc	AveSil	%pSil
1	A	***HCA***	22%	0,85	0,06	0,81	**0,92**	0,35	0,81
2	B	***HCA***	22%	0,85	0,06	0,81	**0,92**	0,35	0,81
3	C	***HCA***	33%	0,93	0,18	0,70	**0,85**	0,27	0,74
4	A	***K-M***	22%	0,83	0,06	0,80	**0,92**	0,34	0,80
5	B	***K-M***	23%	0,85	0,08	0,79	**0,91**	0,35	0,81
6	C	***K-M***	21%	0,80	0,06	0,79	**0,92**	0,37	0,84
7	A	***KMed***	25%	0,93	0,08	0,82	**0,92**	0,31	0,75
8	B	***KMed***	22%	0,90	0,05	0,86	**0,94**	0,34	0,78
9	C	***KMed***	21%	0,83	0,05	0,81	**0,93**	0,36	0,83
10	A	***SOS***	20%	0,72	0,07	0,72	**0,88**	0,33	0,86
11	B	***SOS***	20%	0,73	0,07	0,73	**0,89**	0,34	0,87
12	C	***SOS***	20%	0,80	0,05	0,80	**0,92**	0,37	0,89
13	A	***IF***	23%	0,88	0,07	0,81	**0,92**	0,37	0,86
14	B	***IF***	30%	0,98	0,13	0,78	**0,89**	0,29	0,78
15	C	***IF***	20%	0,85	0,04	0,85	**0,94**	0,39	0,89
16	A	***EE***	10%	0,50	0,00	0,67	**0,90**	0,52	0,96
17	B	***EE***	15%	0,58	0,04	0,66	**0,88**	0,48	0,95
18	C	***EE***	20%	0,60	0,10	0,60	**0,84**	0,43	0,92
19	A	***kNN***	15%	0,58	0,04	0,66	**0,88**	0,5	0,97
20	B	***kNN***	20%	0,6	0,10	0,60	**0,84**	0,43	0,92
21	C	***kNN***	25%	0,68	0,14	0,60	**0,82**	0,38	0,89
**MinMaxDiff*			*13%*	*0,25*	*0,14*	*0,16*	*0,09*	*0,12*	*0,15*

For classification quality measures Sensitivity (Sen), F1 score (F1), and Accuracy (Acc), the higher the value, the better the results. For the Fall-out (FPR) measure, in the classic scenario, the lower the value, the better. However, in the case studied on the basis of financial data, it is interesting to note that a fall-out hit might be a suggestion for experts in the financial department to have a closer look at such a case and evaluate it again.

The results of the kNN algorithm as a simple baseline algorithm are given only for comparison purposes. The results of the kNN algorithm's classification measures are weaker than those of the other methods presented dedicated to solving anomaly detection problems. To assess the similarity of the clustering algorithms (using the author's DACA script) and anomaly detection, the weakest EE and kNN algorithms were omitted from consideration (the differences in values for the other algorithms are recorded in the last row of Table 1 named *MinMaxDiff). For the key measure of Accuracy (Acc), the largest difference is only 0.09 values between the indicated variants.

The best algorithm according to the classification quality measures Sensitivity (Sen), F1 score (F1), and Accuracy (Acc) appeared to be K-Medoids (in association with DACA). The other algorithm variants are also comparably in the range of differences from 0.09 for the mentioned early Accuracy (Acc) to 0.25 for Sensitivity (Sen).

The plot below (Fig. 1) shows the graphical difference of the selected measures: Sensitivity (Sen), Accuracy (Acc), Percentage of Positive Silhouette (% Positive Silhouette - %pSil) for the best algorithm variants (excluding the last EE algorithm).

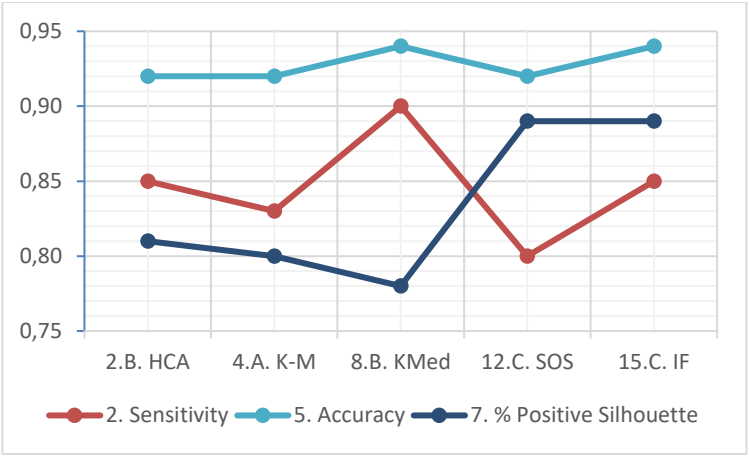

Fig. 1. Selected Results of Quality Measures

Clustering algorithms have the highest Sensitivity (Sen). It is interesting to note that in the case of clusterization measures in Percentage of Positive Silhouette (% Positive Silhouette - %pSil), it is the anomaly detection algorithms (IF, SOS) that achieved the best result. This suggests that it is not the maximum value of the clustering measure that is most important but the appropriate level of it. It should be stressed that there is no perfect correlation between classification and clustering measures.

Finally, the sensitivity of the algorithms to their parameter settings should be mentioned. All methods, both for clustering and dedicated anomaly detection algorithms, depend on the values of initialisation parameters (ap - anomaly percentage and c - contamination). These parameters are crucial in determining the percentage (target) number of anomalies in the set. Therefore, it is important to learn in more detail about the characteristics of the examined collection in order to adjust their value appropriately (e.g., by means of multiple experiments, external evaluation of the results, cross-validation). If the values of these parameters are too low, there is a possibility of not detecting all anomalies (low

true-positive, low sensitivity - recall), if they are too high, there is a possibility of misjudging too many normal observations as anomalies (high false-positive). Moreover, in the case of the clustering algorithm, the parameter determining the target number of clusters appears to be the most essential. As above, the parameter should be chosen appropriately (it is worth using silhouettes, Elbow method). Too small a number of clusters causes the blurring of relevant information, simplifies the structure of the data, consequently, makes it impossible to detect all anomalies. Too many clusters cause overfitting the model, overfitting to a specific situation, which consequently introduces at a later stage misclassification of anomalies.

6. Conclusions and future work

The application of the own script ADAC (Determine Abnormal Clusters Algorithm) for clustering algorithms allows one to achieve comparable or better results than dedicated algorithms for the problem of anomaly detection, based on the instance of specific financial data. Unsupervised machine learning methods (including those based on clustering) for financial anomaly detection are able to support the work of financial professionals responsible for analysing asset and financial data. The development and exploitation of such tools is intended to provide assistance in detecting and identifying deficiencies, fraud, abuse, misconduct, and crimes including financial ones. The identified anomalies should be analysed from multiple perspectives by the aforementioned experts.

On the basis of the algorithms presented, an ensemble method may be elaborated. In such a solution, the sum of the results will permit one to determine the strength of the anomalies, thus the priorities for analysis. Defining a weighting policy for individual algorithms can increase the effectiveness of this method. Finally, it is possible to extend such a method with additional algorithms and their parameter variants. An ensemble method constructed from multiple algorithms and their parameter variants can allow the benefits of multiple methods (e.g. nature of construction, mode of operation, application) for both clustering and dedicated anomaly detection algorithms to be utilised. This paper is an introduction to the construction of such a tool.

References

[1] ANGIULLI F., CLARA P., *Fast Outlier Detection in High Dimensional Spaces*, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 2431, 2002, pp. 15–27.

[2] BERKHIN P., *A survey of clustering data mining techniques*, Grouping Multidimensional Data: Recent Advances in Clustering, 2006, pp. 25-71.

[3] CHMIELEWSKI, M., et al., *Military and Crisis Management Decision Support Tools for Situation Awareness Development Using Sensor Data Fusion*, Advances in Intelligent Systems and Computing, 656, 2018, pp. 189–199.

[4] CZERNIEC I., *Oświadczenia majątkowe. Polska*, Przegląd antykorupcyjny czasopismo Centralnego Biura Antykorupcyjnego, Centralne Biuro Antykorupcyjne, 1, 2019, pp. 53-77.

[5] EZE Peter U., et al., *Anomaly Detection in Endemic Disease Surveillance Data Using Machine Learning Techniques*, Healthcare (Basel), vol. 11(13), 2023, p. 1896.

[6] FIJAŁKOWSKA J., *Fałszowanie informacji ekonomiczno-finansowej w sprawozdawczości przedsiębiorstw*, Etyka w służbie biznesu, Studia i Monografie, 44, 2013, 111- 121.

[7] HE Z., et al. *Discovering Cluster-Based Local Outliers*, Pattern Recognition Letters, vol. 24, no. 9–10, 2003, pp. 1641–1650.

[8] JANSSENS J.H.M., HUSZÁR F., POSTMA E., *Stochastic outlier selection*, Technical Report, Technical report TiCC TR, Tilburg University, vol 1, 2012.

[9] JOHN H., NAAZ S., *Credit Card Fraud Detection Using Local Outlier Factor and Isolation Forest*, International Journal of Computer Sciences and Engineering, vol. 7, no. 4, 2019, pp. 1060–1064.

[10] JOHNSON S. C, Hierarchical clustering schemes. Psychometrika, 32, 1967, 241–254.

[11] JUN S., *An Ensemble Method for Validation of Cluster Analysis*, International Journal of Computer Science Issues (IJCSI), vol 8(6), 2011, pp. 26-30.

[12] KAUFMAN L., ROUSSEEUW P., *Clustering by means of medoids*, In Statistical Data Analysis Based on the L1-Norm and Related Methods, 1987, pp. 405-416.

[13] KAUFMAN L., ROUSSEEUW P., *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley, 1990.

[14] KONOPKA E., PELIKANT A., *Zastosowanie metod grupowania w analizie sieci społecznościowych*, Zeszyty Naukowe WSInf, vol. 13(1), 2014, pp. 13-37.

[15] KUTERA M., *Audyt finansowy, a przestępstwa gospodarcze*, Zeszyty Teoretyczne Rachunkowości, 105(49), 2009, pp 109-121.

[16] LIU F. T., TING K. M., ZHOU Z.-H., *Isolation forest*, In Proceedings of the 2008 Eighth IEEE International Conference on Data Minin, IEEE Computer Society, 1963, pp. 413-422.

[17] LLOYD S. *Least Squares Quantization in PCM*, IEEE Transactions on Information Theory, vol. 28(2) 1982, pp. 129–137.

[18] MACQUEEN J. B., *Some methods for classification and analysis of multivariate observations*, Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1, 1967, pp. 281-297.

[19] MICHERDA B., SZULC M., *Analiza finansowa w badaniu możliwości popełnienia oszustw*, Zeszyty Naukowe Uniwersytetu Ekonomicznego w Krakowie, 785, 2008, pp. 21-31.

[20] NAJGEBAUER A, et al., *Quantitative Methods of Strategic Planning Support: Defending the Front Line in Europe*, Advances in Intelligent Systems and Computing, vol. 656, 2018, pp. 290–299.

[21] PARK H.-S., JUN C.-H., *A simple and fast algorithm for k-medoids clustering*, Expert Systems with Applications, vol. 36(2, part 2), 2009, pp. 3336–3341.

[22] ROUSSEEUW PJ, VAN DRIESSEN K., *A Fast Algorithm for the Minimum Covariance Determinant Estimator*, Technometrics, vol. 41(3), 1999, pp. 212–223.

[23] STOJANOVIĆ B., et al., *Follow the Trail: Machine Learning for Fraud Detection in Fintech Applications*, Sensors (Basel, Switzerland), vol. 21(5), 2021, pp. 1–4.

[24] WANG R., et al. *Local Dynamic Neighborhood Based Outlier Detection Approach and Its Framework for Large-Scale Datasets*, Egyptian Informatics Journal, vol. 22, no. 2, 2021, pp. 125–132.

[25] WARD J., *Hierarchical Grouping to Optimize an Objective Function*, Journal of the American Statistical Association, vol 58(301), 1963, pp. 236-244.

Nienadzorowane uczenie maszynowe w wykrywaniu anomalii finansowych: algorytmy klasteryzacji a metody dedykowane

STRESZCZENIE: Artykuł przedstawia zastosowanie wybranych algorytmów klasteryzacji do wykrywania anomalii w danych finansowych w porównaniu do kilku dedykowanych algorytmów dla tego problemu. W celu wykorzystania algorytmów klasteryzacji do wykrywania anomalii opracowano i zaimplementowano Determine Abnormal Clusters Algorithm (DACA). Ten sparametryzowany skrypt umożliwia na automatyczne wykrycie klastrów zawierających anomalie, na podstawie zdefiniowanych miar odległości. Takie działanie pozwala na szybkie i skuteczne dostosowanie algorytmów klasteryzacji do wyszukiwania anomalii. Przygotowane środowisko badawcze pozwoliło na porównanie wybranych algorytmów klasteryzacji: Hierarchical Cluster Analysis, K-Means, K-Medoids oraz wykrywania anomalii: Stochastic Outlier Selection, Isolation Forest, Elliptic Envelope, Badania przeprowadzono na rzeczywistych danych finansowych, w szczególności dotyczących dochodów zadeklarowanych w oświadczeniach majątkowych wybranej grupy zawodowej. Wykorzystano doświadczenie ekspertów finansowych do oceny anomalii. Ponadto, wyniki oceniono na podstawie wielu popularnych miar klasyfikacji i klasteryzacji. Najlepsze wyniki dla badanego problemu finansowego przedstawił algorytm K-Medoids w połączeniu ze skryptem DACA. W przyszłości warto przebadać metody złożone oparte o przedstawione rozwiązanie.

SŁOWA KLUCZOWE: wykrywanie anomalii, klasteryzacja, uczenie maszynowe, oszustwo finasowe, finase

Received by the editorial staff on: 18.09.2023