Outlier mining in high-dimensional data using the Jensen–Shannon divergence and graph structure analysis

Journal of Physics: Complexity

PAPER

Outlier mining in high-dimensional data using the Jensen–Shannon divergence and graph structure analysis

Alex S O Toledo[1,2], Riccardo Silini[1], Laura C Carpi[3,4] and Cristina Masoller[1,*]

[1] Departament de Fisica, Universitat Politecnica de Catalunya, Rambla Sant Nebridi 22, 08222 Terrassa, Barcelona, Spain
[2] Programa de Pós-Graduação em Modelagem Matemática e Computacional, Centro Federal de Educação Tecnológica de Minas Gerais, Av. Amazonas, 7675, 30510-000, Nova Gameleira, Belo Horizonte, Minas Gerais, Brazil
[3] Instituto Nacional de Ciencia e Tecnologia de Sistemas Complexos (INCT SC), Belo Horizonte, Brazil
[4] Machine Intelligence and Data Science Laboratory (Minds), Departamento de Engenharia Eletrica, Universidade Federal de Minas Gerais, Av. Pres. Antônio Carlos, 6627, 31270-901, Pampulha, Belo Horizonte, Minas Gerais, Brazil
* Author to whom any correspondence should be addressed.

E-mail: cristina.masoller@upc.edu

Keywords: anomaly detection, outlier detection, graphs, percolation, Jensen–Shannon divergence

OPEN ACCESS

RECEIVED
30 June 2022

REVISED
29 November 2022

ACCEPTED FOR PUBLICATION
6 December 2022

PUBLISHED
15 December 2022

Abstract

Reliable anomaly/outlier detection algorithms have practical applications in many fields. For instance, anomaly detection allows to filter and clean the data used to train machine learning algorithms, improving their performance. However, outlier mining is challenging when the data is high-dimensional, and different approaches have been proposed for different types of data (temporal, spatial, network, etc). Here we propose a methodology to mine outliers in generic datasets in which it is possible to define a meaningful distance between elements of the dataset. The methodology is based on defining a fully connected, undirected graph, where the nodes are the elements of the dataset and the links have weights that are the distances between the nodes. Outlier scores are defined by analyzing the structure of the graph, in particular, by using the Jensen–Shannon (JS) divergence to compare the distributions of weights of different nodes. We demonstrate the method using a publicly available database of credit-card transactions, where some of the transactions are labeled as frauds. We compare with the performance obtained when using Euclidean distances and graph percolation, and show that the JS divergence leads to performance improvement, but increases the computational cost.

1. Introduction

The terms anomaly and outlier generally refer to observations of a give process that seems to be generated by a mechanism that is not the one that governs the process. While outliers and anomalies are often used as synonymous, they have also been distinguished in the following terms: an outlier is a legitimate data point that is far from the center of a distribution that characterizes a process, and an anomaly is a data point that cannot be explained given current knowledge of the process generating the data. Anomalies have been classified in three types: point anomalies (a data point that is anomalous with respect to the rest of the data), contextual anomalies (a data point that is anomalous in a specific context) and collective anomalies (a set of data points that are not anomalies by themselves, but their collective occurrence is anomalous) [1].

The identification of these observations is important for different purposes (failure detection, novelty detection, intruder detection, etc), and in particular, in the context of artificial intelligence systems, to clean the data used for training the algorithms.

Many methods for outlier detection have been proposed in the literature (see, e.g. [2–6] and references therein), some of them, based on distances that can be computed between elements of the dataset [7–11]. In outlier detection via graph methods, distance-based outlier mining is based on a fully connected graph structure in which the nodes represent the elements of the dataset and the connections between them are quantified by a distance measure. In this sense, with N elements in the dataset, each with a M-dimensional vector of features, $f_i = \{f_{i1}, f_{i2}, \ldots f_{iM}\}$ $i = 1 \ldots N$, by using an appropriate distance to quantify differences

in the vectors of features, one obtains a $N \times N$ distance matrix, where the vector of distances of node i, $d_i = \{d_{i1}, d_{i2}, \ldots d_{iN}\}$, can be considered a new set of features that can be used for training a binary classification algorithm (outlier/normal element). In large datasets the new feature vector is very long and this approach becomes computationally demanding. An alternative approach is to use a dimensionality-reduction strategy and extract informative features from the vectors of distances. In this work we show that the average distance and the shape of the distance distribution can be used for outlier mining.

A popular measure for comparing different distributions is the Jensen–Shannon (JS) divergence [12]. We define new outlier scores (OSs) using the JS divergence computed from the distributions of Euclidean distances between nodes. The method does not have any free parameter and thus, it does not require training. We demonstrate the method using a publicly available database of credit-card transactions where some of the transactions are labeled as frauds [13, 14]. We quantify the performance with well-known measures, the area under the curve receiver operating characteristic (AUC-ROC) and the area under the curve precision recall (AUC-PR) [15, 16]. We also compare with the performance of the 'graph-percolation' method proposed in [10], which is also parameter-free. In the percolation method, the links with longest distances are gradually removed, and an OS is assigned to each element, in the order in which the elements become disconnected from the giant component.

The work is organized as follows. Section 2 describes the proposed methodology, section 3 describes the dataset used, section 4 describes the performance quantifiers, section 5 presents the results and section 6 presents the discussion and the conclusions.

2. Method

We consider a set of N elements (nodes) which have associated vectors with M features (for the credit card database to be described in the next section, $M = 28$). The Euclidean distance between two elements, i and j, whose feature vectors are $\{f_{i1} \ldots f_{iM}\}$ and $\{f_{j1} \ldots f_{jM}\}$ is:

$$d_{ij} = \sqrt{\sum_{k=1}^{M} (f_{ik} - f_{jk})^2}. \tag{1}$$

With the vector of distances of node i, $d_i = \{d_{i1}, d_{i2}, \ldots d_{iN}\}$, we define the first 'outlier score' of node i, $\mathrm{OS1}_i$ as the average distance,

$$\mathrm{OS1}_i = \frac{1}{N} \sum_{l=1}^{N} d_{il}. \tag{2}$$

Elements that have high values of OS1 have, on average, large distances to other nodes.

More information can be obtained by inspecting not the average, but the shape of the distribution of distances. Given two nodes i and j, if the shape of the distributions of Euclidean distances $\{d_{il}\}$ and $\{d_{jl}\}$ (with $l = 1 \ldots N$) is similar, then the elements can be considered similar, else, they are different. Therefore, a weight can be assigned to the link between nodes i and j by calculating the distance between the distributions of $\{d_{il}\}$ and $\{d_{jl}\}$ values, P_i and P_j respectively. Different distance measures can be used to compare two distributions and here we use the popular JS divergence,

$$D_{ij} = \mathrm{JS}[P_i, P_j] = H[(P_i + P_j)/2] - H[P_i]/2 - H[P_j]/2, \tag{3}$$

where the Shannon entropy, $H[P]$, is defined as follows [17]: given a discrete random variable X which takes values in the alphabet Ξ and is distributed according to $p : \Xi \to [0, 1]$,

$$H(X) = -\sum_{x \in \Xi} p(x) \log(p(x)). \tag{4}$$

If the distribution is estimated from an histogram with L bins, then $H[P] = -\sum_{m=1}^{L} p_m \log p_m$ with $\sum_{m=1}^{L} p_m = 1$. We have written the algorithm in *python* and used the 'euclidean' library to calculate the Euclidean distances and the 'jensenshannon' library to calculate the JS divergence [18].

Then, with the new vector of distances of node i, $D_i = \{D_{i1}, D_{i2}, \ldots D_{iN}\}$, we repeat the procedure and calculate a second OS of node i as the average distance:

$$\mathrm{OS2}_i = \frac{1}{N} \sum_{l=1}^{N} D_{il}. \tag{5}$$

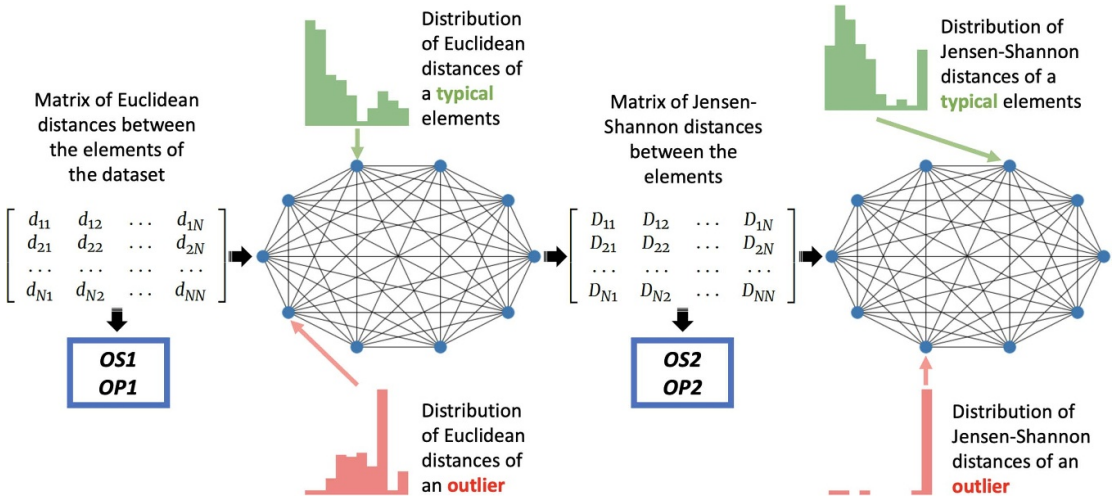

Figure 1. Schematic representation of the procedure for assigning outlier scores (OS) to elements of a database that are represented by network nodes. OS1 and OS2 are defined in equations (2) and (5), OP1 and OP2 refer to the order in which a node becomes disconnected from the giant component, as the longest links are removed one by one, when using Euclidean distances (OP1) or when using JS divergences (OP2).

In addition, we calculate OSs using the graph percolation procedure proposed in [10]. In this procedure, the connections with longest distances are removed one by one, and an OS is assigned to each element, in the order in which the element becomes disconnected from the giant component. For example, if node number 3 is the first node to become disconnected from the giant component, and then nodes 5 and 11 become disconnected, then, node number 3 is assigned an OS OP = 1, node number 5 is assigned OP = 2 and node number 11, OP = 3 (a video showing an example is available in [10], supplementary information). We refer to the OSs defined in this way, as OP1 when using Euclidean distances and as OP2, when using JS divergences.

The whole procedure is schematically summarized in figure 1: we start with the $N \times N$ matrix of Euclidean distances (d). Using this matrix, we define OS1 as the average Euclidean distance to all the other nodes and OP1 is the order in which is disconnected from the giant component. Through the probability distributions obtained from the values of distances for each node, the JS divergence is computed for each pair of nodes, and a new $N \times N$ distance matrix is obtained (D). With this matrix, we define OS2 as the average JS divergence, and OP2 is the order in which the element becomes disconnected from the giant component, when using JS divergences.

3. Data

The dataset used in this study represents 284 807 European credit card transactions carried out in two days in September 2013 [14]. Transactions are anonymous and labeled in two classes: fraud (1) or non-fraud (0). The dataset is very unbalanced, with 492 frauds and 284 315 non-frauds, where frauds represent only 0.172% of the dataset. The dataset is anonymized by a principal component analysis with 28 dimensions.

The data structure contains 31 numeric attributes. The 'Time' defines the seconds between transactions and the first transaction. The 'Amount' defines the value associated with each transaction. The 'Class' attribute specifies the prevision of transactions, where the value 1 represents fraud, and the value 0 represents non-fraud. The other 28 attributes have values taken from PCA.

In [10] no improvement in average precision was found when including the attributes 'Time' and 'Amount' in the feature set and therefore, here we only use the 28 PCA features in the analysis.

4. Performance measures

As important as selecting data attributes (features) to mine outliers is selecting an appropriate performance measure. Two well-known measures are the AUC-ROC and the AUC-PR [15, 16].

The curves are obtained by varying the classification threshold and calculating the

(a) True positives (TP): number of elements that are correctly detected as outliers.
(b) True negatives (TN): number of elements that are correctly detected as normal.
(c) False positives (FP): number of elements that are incorrectly detected as outliers.
(d) False negatives (FN): number of elements that are incorrectly detected as normal.

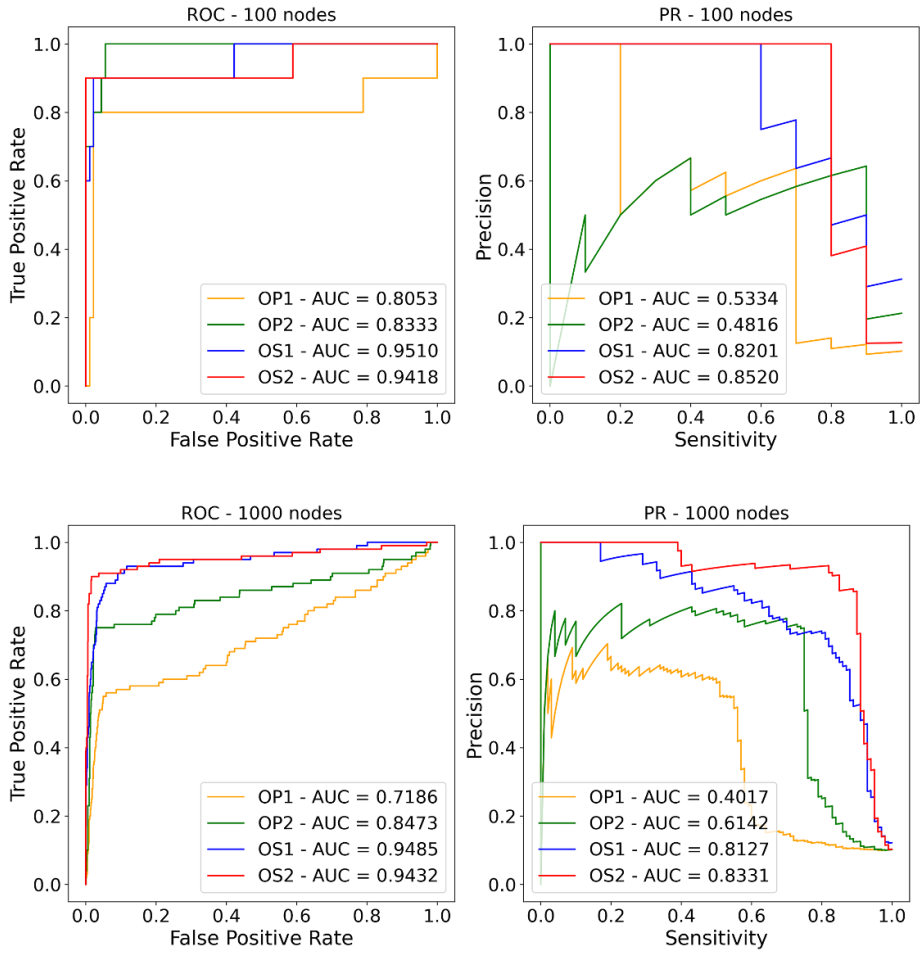

Figure 2. Receiver operating characteristic (left column) and precision-recall curve (right column) obtained for the four outlier scores analyzed, using 200 networks with 10% fraud and 90% no fraud. Network sizes: $N = 100$ (top row), $N = 1000$ (bottom row).

Then, the true positive rate (TPR, also known as *recall*) is TP/(TP+FN) = TP/(# of outliers), and the false positive rate (FPR) is FP/(TN+FP) = FP/(# of normal elements). The ROC curve is obtained by plotting the TPR vs. FPR.

The area under the ROC curve is a measure of the goodness of a binary classifier: while random guessing gives a diagonal line, a perfect classifier has one (or more) thresholds that perfectly separate the two classes. In this situation, AUC-ROC = 1.

ROC curves can present a very optimistic view of a classifier performance if there is a significant class unbalance and in this case PR curves have been used as an alternative to ROC curves [15, 16]. The PR curve is more informative because it does not depend on the number of true negatives. The *precision* is the ratio of correct positive detections over all positive detections, TP/(TP+FP), and the PR curve is obtained by plotting the precision vs. the recall (i.e. the TPR). The *average precision* is the area under the PR curve (AUC-PR).

5. Results

Figure 2 displays the ROC and PR curves obtained with the four OSs, considering datasets with different sizes. In all cases, the elements of the datasets were selected such that 10% were fraud transactions and 90% were regular ones. We note that OS1 and OS2 have very similar performance, which is higher than the performance of OP1 and OP2. In [10] the average performance achieved with the percolation method was lower than 0.6, while here, the performance of OS1 and OS2 typically exceed 0.8. We have verified that the value of AUC-PR remains similar when a smaller number of outliers is included in the dataset (figure 3).

Results with different N are summarized in table 1, where it can be seen that OS1 and OS2 clearly outperform the percolation-based methods, OP1 and OP2. However, one might wonder if this increase in performance does not require a high increase of the computational time. Figure 4 shows that, as could be expected, for the four methods the computational time increases as $\propto N^2$ due to the calculation of the $N \times N$

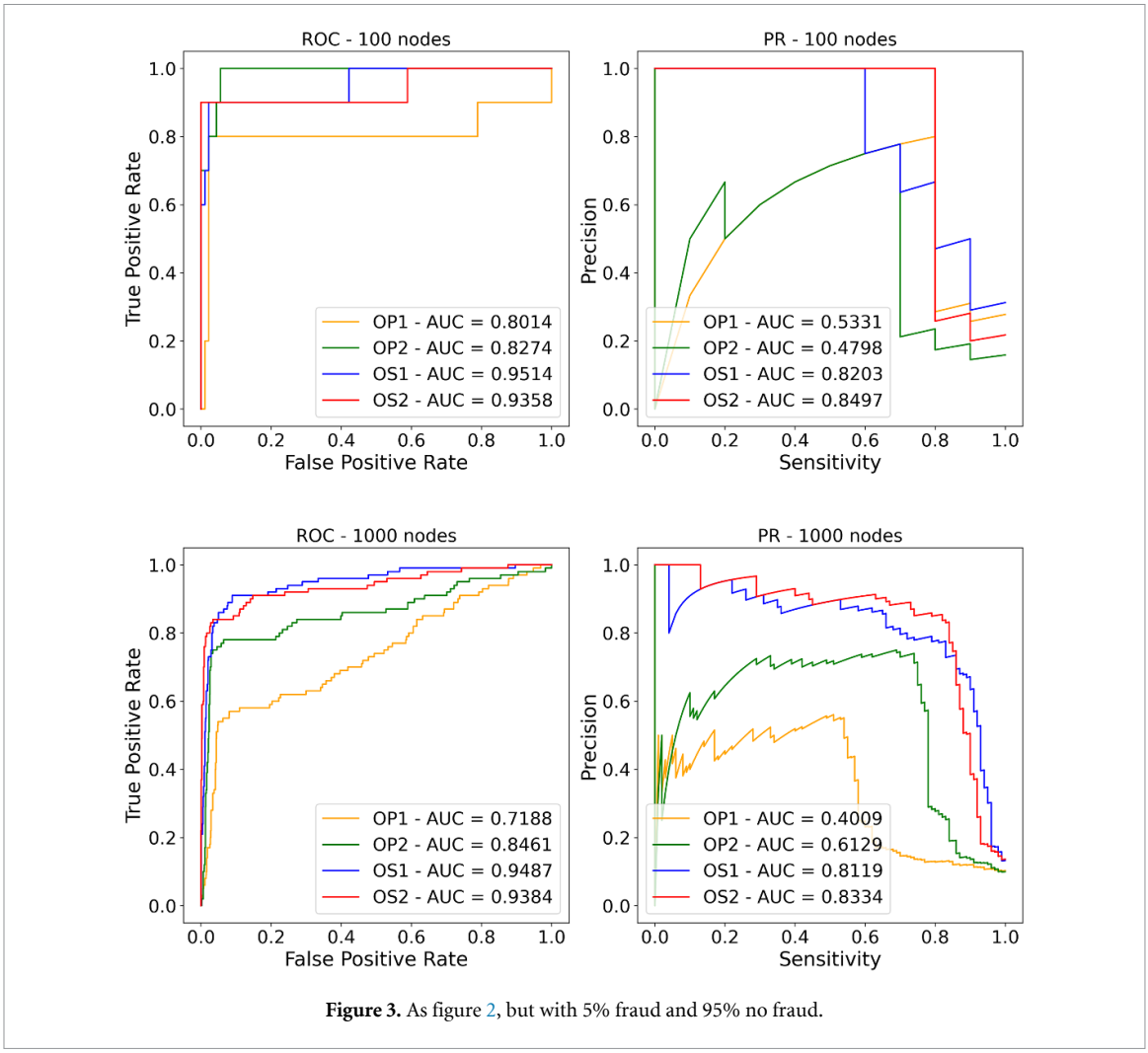

Figure 3. As figure 2, but with 5% fraud and 95% no fraud.

Table 1. Performance obtained for datasets of different sizes, *N*. For each *N* the mean and the standard deviation of the AUC-PR were calculated from 200 datasets composed by different elements, such that 90% are normal transactions and 10% are frauds.

	OP1		OP2		OS1		OS2	
N	mean	std	mean	std	mean	std	mean	std
100	0.53	0.17	0.48	0.11	0.82	0.11	0.85	0.10
200	0.48	0.12	0.55	0.12	0.82	0.08	0.85	0.07
500	0.41	0.08	0.61	0.08	0.82	0.05	0.84	0.05
1000	0.40	0.07	0.61	0.07	0.81	0.04	0.83	0.04
5000	0.38	0.04	0.62	0.04	0.81	0.02	0.82	0.02
10 000	0.37	0.03	0.62	0.03	0.82	0.02	0.83	0.02

distance matrix. Figure 4 indicates some typical times when the algorithm was run on a iMac core i7 computer with 64 GB RAM.

In table 1 we also notice that for OP1, the mean of the AUC-PR seems to decrease and then seems to saturate as *N* increases. We do not yet know the origin of this variation, and to determine whether is generic or is specific to the dataset analyzed, additional studies, using other datasets, are planned.

Taken together, the results presented in table 1 and figure 4 show that OS1 is a cost-effective method for mining outliers in this dataset, because it has a performance that is only slightly lower than OS2, while it avoids the calculation of the matrix of JS divergences, which is computationally demanding when the datasets are large.

We remark that detecting credit card frauds is an active research field, and the publicly available dataset that we have used [13, 14], has been used by other authors. A natural next step is to perform a critical comparison with other techniques; however, this is left for future work as our goal here is to present a general method that can be used to mine outliers in generic datasets, whenever appropriate distances can be defined

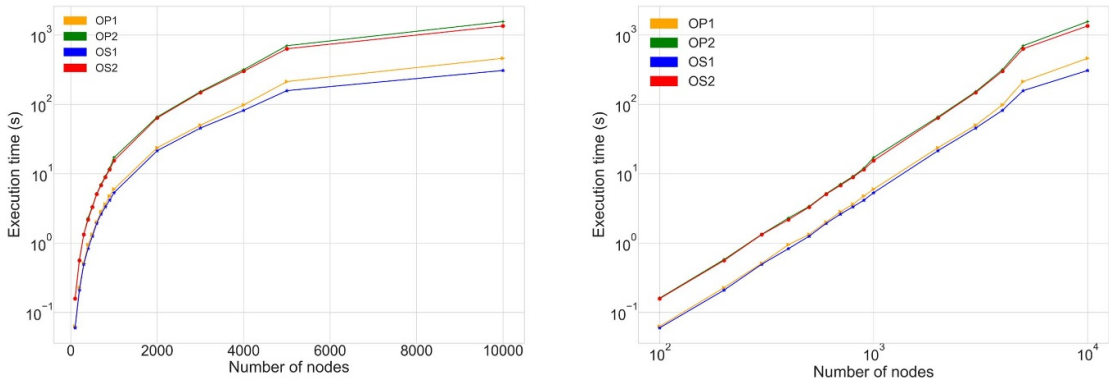

Figure 4. Computational time as a function of the network size. The linear relation is consistent with the fact that the number of operations grows a $\propto N^2$. Methods OS2 and OP2 require consistently more time, as the JS matrix needs to be calculated from the Euclidean matrix.

between elements of the datasets. We use the credit card database just as an example, and we do not claim that our method outperforms other methods for detecting credit card frauds.

6. Conclusion

We have proposed a new methodology for mining outliers in high-dimensional datasets. The method is based in the analysis of the fully connected graph, where the distances between the elements of the dataset are defined by using the Euclidean or the JS divergence. At least for the dataset analyzed we conclude that the analysis of the average distance improves precision, with respect to the graph percolation method proposed in [10]. Future work will be devoted to test these methods in databases of crime occurrences in Minas Gerais, Brazil [19].

As the JS divergence complements the information obtained from the mean Euclidean distances (because it compares the *shape* of two distributions), we speculate that combining OS1 and OS2 can further improve performance. Another way in which the method can be refined is by replacing the (parameter-free) Euclidean distance with a functional form whose parameters can be optimally tuned to the data that is analyzed. The method proposed here is parameter-free and therefore, the algorithm does not require any training; however, a functional form can be used instead of the Euclidean distance, and its parameters can be optimally tuned to the particular data under inspection, by using machine learning. For example, weights can be assigned to different features, to pay particular attention to the most informative ones.

Data availability statement

The dataset used in this study can be downloaded in [14]. The data that support the findings of this study are openly available at the following URL/DOI: www.kaggle.com/datasets/mlg-ulb/creditcardfraud.

Acknowledgment

R S was funded by MSCA Innovative Training Network Climate Advanced Forecasting of sub-seasonal Extremes (CAFE H2020-813844); C M was funded by Project PID2021-123994NB-C21 of Ministerio de Ciencia e Innovacion, Spain, and the ICREA ACADEMIA program of Generalitat de Catalunya.

Declarations

Conflicts of interest
The authors declare no competing interests.

Research involving human participants and/or animals
The research did not involve humans or animals.

ORCID iDs

Alex S O Toledo https://orcid.org/0000-0001-8413-5524
Cristina Masoller https://orcid.org/0000-0003-0768-2019

References

[1] Chandola V, Banerjee A and Kumar V 2009 Anomaly detection: a survey *ACM Comput. Surv.* **41** 1–58
[2] Akoglu L, Tong H and Koutra D 2015 Graph based anomaly detection and description: a survey *Data Min. Knowl. Discov.* **29** 626–88
[3] Wang H, Bah M J and Hammad M 2019 Progress in outlier detection techniques: a survey *IEEE Access* **7** 107964–8000
[4] Boukerche A, Zheng L and Alfandi O 2020 Outlier detection: methods, models and classification *ACM Comput. Surv.* **53** 1–37
[5] Pang G, Shen C, Cao L and Hengel A V D 2021 Deep learning for anomaly detection: a review *ACM Comput. Surv.* **54** 1–38
[6] Blázquez-García A *et al* 2021 A review on outlier/anomaly detection in time series data *ACM Comput. Surv.* **54** 1–33
[7] Ramaswamy S, Rastogi R and Shim K 2000 Efficient algorithms for mining outliers from large data sets *Proc. 2000 ACM SIGMOD Int. Conf. on Management of Data*
[8] Angiulli F, Basta S and Pizzuti C 2005 Distance-based detection and prediction of outliers *IEEE Trans. Knowl. Data Eng.* **18** 145–60
[9] Radovanović Mš, Nanopoulos A and Mirjana I 2014 Reverse nearest neighbors in unsupervised distance-based outlier detection *IEEE Trans. Knowl. Data Eng.* **27** 1369–82
[10] Amil P, Almeira N and Masoller C 2019 Outlier mining methods based on graph structure analysis *Front. Phys.* **7** 194
[11] Erz M *et al* 2021 Anomaly detection in multidimensional time series—a graph-based approach *J. Phys. Complex.* **2** 045018
[12] Thomas M T C A J and Thomas Joy A 2006 *Elements of Information Theory* (New York: Interscience)
[13] Pozzolo D, Boracchi G, Caelen O, Alippi C and Bontempi G 2017 Credit card fraud detection: a realistic modeling and a novel learning strategy *EEE Trans. Neural Netw. Learn. Syst.* **29** 3784–97
[14] Credit card fraud detection (available at: www.kaggle.com/datasets/mlg-ulb/creditcardfraud) (Accessed 19 May 2022)
[15] Davis J and Goadrich M 2006 The relationship between precision-recall and ROC curves *Proc. of the 23rd Int. Conf. on Machine Learning*
[16] Saito T and Rehmsmeier M 2015 The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets *PLoS One* **10** e0118432
[17] Shannon C 1948 A mathematical theory of communication *Bell Syst. Tech. J.* **27** 379–423
Shannon C 1948 *Bell Syst. Tech. J.* 27 623–56.
[18] Distance computations (scipy.spatial.distance) (available at: https://docs.scipy.org/doc/scipy/reference/spatial.distance.html) (Accessed 02 April 2022)
[19] Toledo A S O, Carpi L C and Atman A P F 2020 *Diversity Analysis Exposes Unexpected key Roles in Multiplex Crime Networks* (*Complex Networks XI*) (Cham: Springer) pp 371–82