AUTOMATIC CLUSTERING OF INDUSTRIAL DATA WITH THE CONNECTED COMPONENTS

Katarína Firdová, Céline Labart, Laurent Vuillon

AUTOMATIC CLUSTERING OF INDUSTRIAL DATA WITH THE CONNECTED COMPONENTS

KATARÍNA FIRDOVÁ, CÉLINE LABART, AND LAURENT VUILLON

Abstract. This paper proposes the use of techniques from the field of graph theory to automatically determine and cluster the same types of observations without prior knowledge on their number. The idea is to create a graph where observations are vertices and an edge between two observations exists if and only if one observation is in the nearest neighbourhood of another one. Connected components of the graphs can be then considered as groups of observations of the same type. The approach is particularly useful for multiclass anomaly detection in the context of Industry 4.0.

Keywords : Industry 4.0, clustering, connected components

1. Introduction and related work

In the setting of global transformation toward data driven world that we are experiencing in the 21st century, manufacturing sector is changing. Modern plants integrate smart technologies that allow real-time communication between the machines in order to increase the efficiency, automate and change the tools and roles of industrial teams. Correctly used, information obtained from data help to control the process and improve the factory in any way.

Progress in the technology enables the industrials to install devices which collect and store data from multiple sources automatically but their full potential is often left unexplored. Complexity of the collected data is beyond the limits of human perception and special tools are required to facilitate the analysis. Lot of research is focused on development of the methodologies that can deal with unlabelled or unstructured data and valorise them automatically.

The clustering is one of the fundamental strategies of data analysis when unlabelled datasets with different types of observations are examined. It groups similar observations together providing an important insight to the composition of the dataset and simplifying its further processing.

Many clustering algorithms exist, using different notions of similarity for the regrouping procedure (e.g. distance, density, distributions) and requiring different parameters (e.g. number of clusters, maximum distance). The latter mentioned are hindering elements in the analysis. Not only it needs to be adjusted for every

Date: March 19, 2024.

dataset, but often this information is missing (e.g. when we have no a priori insight about dataset). Moreover, some parameters are unintuitive to tune.

In the context of Industry 4.0 it is desirable to have a clustering method that do not require these parameters. This paper aims to generalise clustering procedure and propose a method which can automatically determine clusters in data related to the industrial processes without customised settings.

Algorithms like K-means [Ste57] and Agglomerative Clustering [agg] which are often applied to cluster together the same types of observations require information about the number of different clusters n_{clust}.

The Silhouette score [Rou87] or Calinski and Harabasz score [CH74] can evaluate the quality of the structures when n_{clust} varies but the final choice of n_{clust} should be preceded by a visual analysis of the corresponding graph.

Density based (DBSCAN [EKSX96]) or Hierarchical density-based (HDBSCAN [CMS13]) clustering algorithms are alternative methods which can determine the number of clusters n_{clust} but require to set a maximum distance of neighbourhood (which is not an intuitive parameter) and/or minimal sample size parameters. Besides, observations which are not in a dense region can *fell out of a cluster*.

In the field of graph theory, communities are parts of the graph with few ties with the rest of the system [For10]. If the set of nodes is internally densely connected, it is considered to some extent as a separated entity (i.e. cluster), although a sparse connection with another community exists. One of the popular techniques for community detection is the modularity maximisation Louvain method [BGLL08]. This one and other similar techniques can be also employed as clustering methods which do not require to precise the n_{clust} parameter, however they need to tune other non-intuitive parameters like modularity gain threshold or resolution parameter.

2. Methodology

Graph data structures find many applications (see [Bar02] or [Bar16]) such as in transportation systems (search for the shortest path between two points) or in social networks (suggestions of people you may know based on mutual friends), but direct application in the clustering when analysing a dataset is less common.

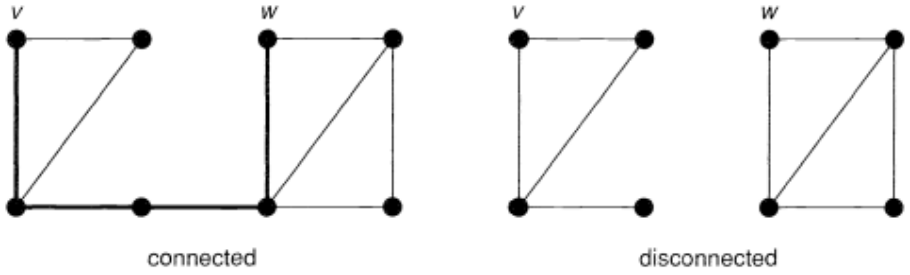
Motivated by the graph structures and their potential, we propose to examine an approach of completely separated subgraphs as individual clusters. The approach consists of determining tuples of nearest neighbours based on their distances and

creating a graph, where one connected component represents one type of observations, i.e. one cluster.

Following the notations used in graph theory [Wil96], a simple graph G consists of a non-empty finite set of elements $V(G)$ named *vertices* (or *nodes*) and a finite set of unordered pairs of elements $E(G)$ named *edges*.

A graph is connected if there exists a finite sequence of distinct edges between each distinct pair of vertices. Examples of connected and disconnected graphs can be seen in Figure 1. Any disconnected graph G can be described as the union of connected sub-graphs, each of which is a component of G.

Figure 1. Example of connected (*in one piece*) and disconnected graphs [Wil96].



Inspired by connected structures, we propose to create a graph where the vertices are the observations. A pair of two observations creates an edge if at least one observation in the pair contains the other in its nearest neighbourhood.

Formally, let $NN_{k,i}$ be a set of k_{neighb} nearest neighbours of observation i and S be a symmetric binary similarity matrix, i.e. $S_{i,j} = S_{j,i} = 1$ if $i \in NN_{k,j}$ or $j \in NN_{k,i}$; $S_{i,j} = S_{j,i} = 0$ otherwise. An edge $\{i, j\}$ which joins vertices i and j exists if and only if $S_{i,j} = 1$.

The idea is to create a graph where a set of similar observations corresponding to one class represents one connected component of a graph. The number of these components represents the number of different classes.

The method is not completely parameter-free and k_{neighb} has to be defined. Yet it is a comprehensible parameter and does not necessarily need different value for different dataset.

The approach will be later referred to as *connected components clustering*.

3. Results

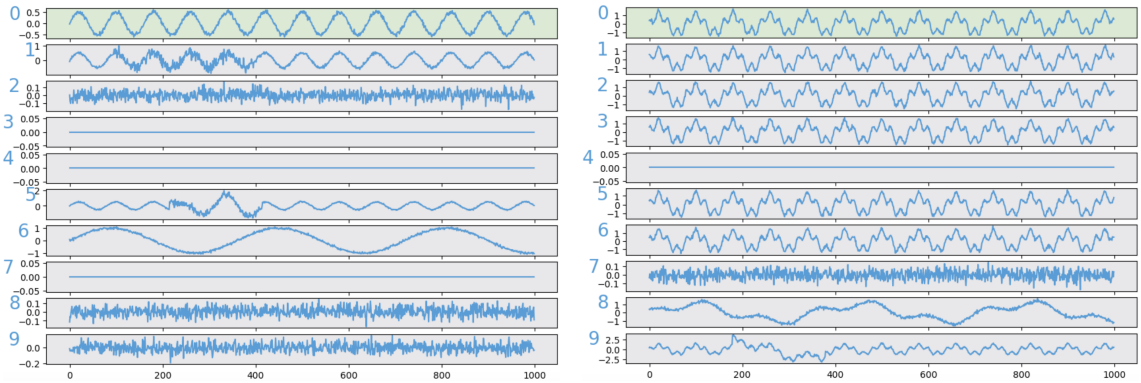
As mentioned above, the main impulse behind this work is to automatise the clustering procedure. In the following we deal with time series multiclass anomaly

detection. We consider global and local anomalies.

The first part of this work is driven by motivation to improve the fluidity of anomaly detection and explaination procedure developed in [CFLM22]. In the mentioned paper, the detected anomalies are to be clustered according to their type to allow the characterisation of each anomaly group. It is an example of a situation when it is useful to group similar observations together (i.e. detected anomalies of the same type) but we may not know in advance how many anomaly types are present.

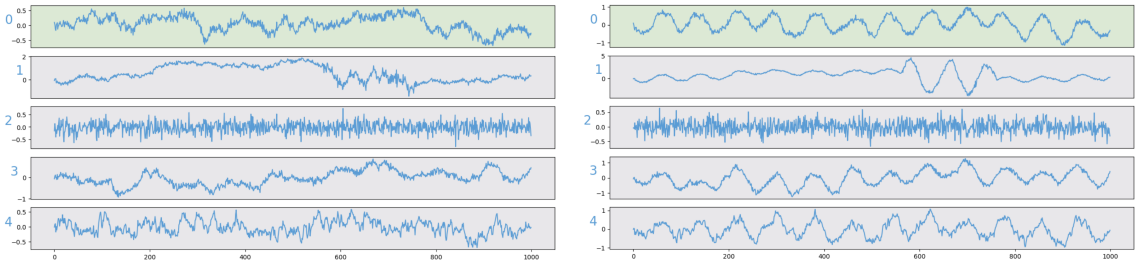
To illustrate the automatised clustering, we use two datasets simulating different types of anomalous continuous processes. One observation, i.e. one anomalous process, is a 2-dimensional chronological series. There are 9 types of anomalies in *Dataset 1* and 4 types in *Dataset 2*. Examples of each type of observation in *Dataset 1* and *Dataset 2* are displayed in Figures 2 and 3, respectively.

We use features vector representation of the processes, i.e. for each chronological series we extract features characterising the behaviour of the corresponding anomalous process. Features are extracted using Python package `tsfresh` [CKLF17].



X_1 X_2

Figure 2. *Dataset 1* contains $N = 500$ of $m = 2$ dimensional time series of length $l = 1000$.
X_1 is represented on the left and X_2 is represented on the right side of the figure. Line 0 (green) is one example of the series with typical behaviour, whereas lines 1 to 9 correspond to different anomalous observations. There are 50 anomalous series in the dataset, composed either of one anomaly on X_1 or of anomalies on both X_1 and X_2.



X_1 X_2

Figure 3. *Dataset 2* contains $N = 520$ of $m = 2$ dimensional time series of length $l = 1000$.
Line 0 (green) corresponds to one series with typical behaviour, whereas lines 1 to 4 correspond to different anomalous observations, displaying X_1 on the left and X_2 on the right side of the graphs. There are 70 anomalous series in the dataset.

Automatically extracted features are numerous, creating highly dimensional vectors of features. This may lead to difficulties when Euclidean distance is used to compute the similarity matrix. We apply Laplacian score [HCN05] to reduce dimensions and to keep only the most relevant features.

Concerning the approach of the *connected components clustering*, the nearest neighbourhood is set as $k_{neighb} = 2$ nearest observations (i.e. vectors of features) in terms of the Euclidean distance.

We can see that the approach of connected components is competitive with other methods using Silhouette score to determine the number of clusters n_{clust} on *Dataset 1* and *2* (Tables 1 and 2 respectively).

Figure 4 illustrates a graph of anomalies from *Dataset 2* as an example. Different colours of the disks (observations) present different connected components, corresponding to the types of anomalies. It is easy to verify visually that the separation is correct in this case, given that indexes (values written in the disk) are coherent.[1]

[1]Indeed, the anomaly groups in *Dataset 2* are generated in ordered way, grouping 1st type of anomalies on first 20 positions (i.e. indexes 0-19), 2nd type of anomalies on following 20 positions (indexes 20-39), 3rd type of anomalies on following 10 positions (indexes 40-49) and 4th type of anomalies on remaining 20 positions (indexes 50-69).

Figure 4. Graph generated in the visualisation software Gephi displaying connected components among anomalies from *Dataset 2*. Laplacian score is used in this case to select the relevant features.

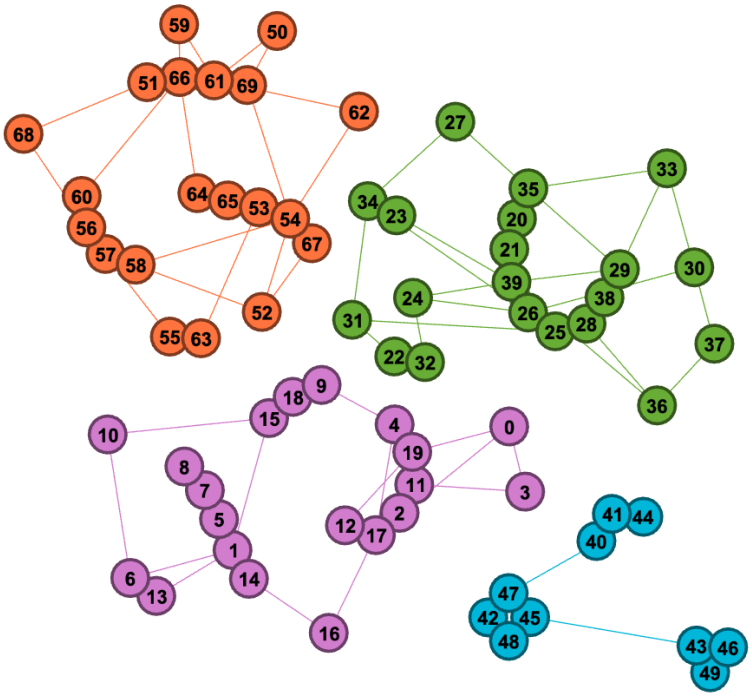


Table 1. Comparison of the performance of selected clustering methods on true anomalies for *Dataset 1*. For K-means and Agglomerative Clustering, Silhouette score is used to determine the number of clusters (n_{clust}). Average Normalised Mutual Information [pytb] and Accuracy scores [pyta] on 5 executions are displayed.

Performance on true anomalies from *Dataset 1* (true $n_{clust} = 9$)						
Feature selection	All features			Laplacian		
Eval. metric	n_{clust}	NMI	ACC	n_{clust}	NMI	ACC
K-means	7	0.91	0.79	7	0.91	0.8
Agglomerative Clustering	7	**0.92**	**0.8**	7	0.91	0.8
Connected components	5	0.77	0.6	8	**0.96**	**0.9**

Table 2. Comparison of the performance of selected clustering methods on true anomalies from *Dataset 2*. For K-means and Agglomerative Clustering, Silhouette score is used to determine the number of clusters (n_{clust}). Average Normalised Mutual Information [pytb] and Accuracy scores [pyta] on 5 executions are displayed.

Performance on true anomalies from *Dataset 2* (true $n_{clust} = 4$)						
Feature selection	All features			Laplacian		
Eval. metric	n_{clust}	NMI	ACC	n_{clust}	NMI	ACC
K-means	2	0.61	0.57	4	1	1
Agglomerative Clustering	2	0.61	0.57	4	1	1
Connected components	2	0.61	0.57	4	1	1

The following real data example deals with clustering of human motion measurements. Dataset [MCCH19] includes chronological data from accelerometer and gyroscope sensors, collected from the users' smartphones. Series are transformed to vectors of features and *connected components clustering* is applied to identify and to group together similar activities performed by users: standing, sitting, upstairs, downstairs, walking and jogging.

Table 3 presents the results. We can observe that only 2 and 3 clusters are detected using all features and selected features, respectively. However, the similar nature of observations in the same group is preserved. Indeed, two clusters are detected when considering all features. They correspond to clusters of static activities (standing and sitting) and dynamic activites (the others). Three clusters are detected after feature selection. They correspond to static (standing and sitting), slow (upstairs, downstairs, walking) and fast activities (jogging). From this point of view, the clustering leads to the correct results. The best results are obtained with K-means, but other clustering methods stay competitive.

Table 3. Comparison of the performance of selected clustering methods on *Human motion* dataset. For K-means and Agglomerative Clustering, Silhouette score is used to determine the number of clusters (n_{clust}). Average Normalised Mutual Information [pytb] and Accuracy scores [pyta] on 5 executions are displayed.

Performance of clustering on *Human motion* dataset.						
Feature selection	All features			Laplacian		
Eval. metric	n_{clust}	NMI	ACC	n_{clust}	NMI	ACC
K-means	2	1	1	3	**0.934**	**0.986**
Agglomerative Clustering	2	1	1	3	0.879	0.968
Connected components	2	1	1	3	0.879	0.968

In addition, we illustrate the results on an example motivated by the clustering of local anomalies on one continuous process.

Figure 5 shows 5 three-dimensional series, considered as individual datasets, i.e. each series is analysed separately. Before the transformation to the vectors of features, each continuous series is divided on rolling windows of size 400 and step 50

values, creating partially overlapping, not homogeneous subseries. Feature extraction is done on each window.

In this demonstration, we aim to cluster local anomalies (red elipses in Figure 5), i.e. rolling windows that partially contain anomaly.

As illustrated in Figure 6, rolling windows on different positions - and therefore corresponding vectors of features - may differ a lot for the same anomaly type. Since the original data come from a time series, we can assume that consecutive positions of rolling windows are supposed to represent the same type of anomaly. If the clustering algorithm detects more clusters on the same anomalous period, we select the final label as a majority vote. This approach enables us to evaluate the capacity to recognise different types of observations, minimising the error caused by heterogenous rolling windows.

Results in Table 4 show that after a feature selection, the method is capable to correctly distinguish different types of observations. However the assumption that different types of anomalies are not overlapping is important. The initial number of identified components is stated in Table 4 as n_{clust}. We can see that the algorithm detects more components i.e. clusters on one continuous period. This situation happens when the ratio of a window containing the anomaly varies during one anomalous period (Figure 6). The performance presented in the following tables is evaluated once different labels on a same anomalous period are merged.

Figure 5. Overview of simulated datasets. Five multidimensional series (labelled 1 - 5) with different anomalies (red ellipses) are generated to test the process of anomaly detection. In this part, we use the datasets to perform the clustering on anomalous rolling windows of size (400,50). Note that different anomalies are not overlapping.

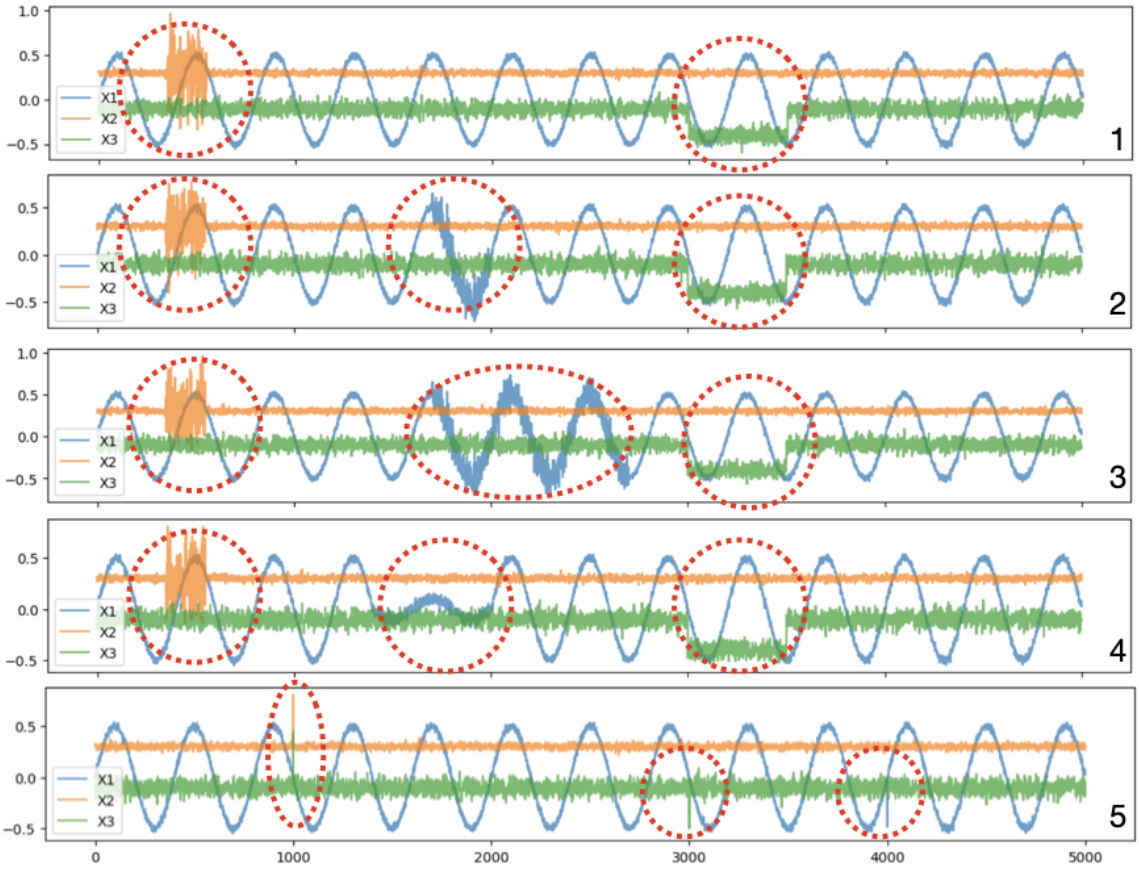


Figure 6. Examples of four different positions of a rolling window on the same anomalous period. The ratio of anomalous values varies from one line to another: the ratio of anomalous values is lower in lines 1 and 4 than in lines 2 and 3, leading to heterogenous subseries.

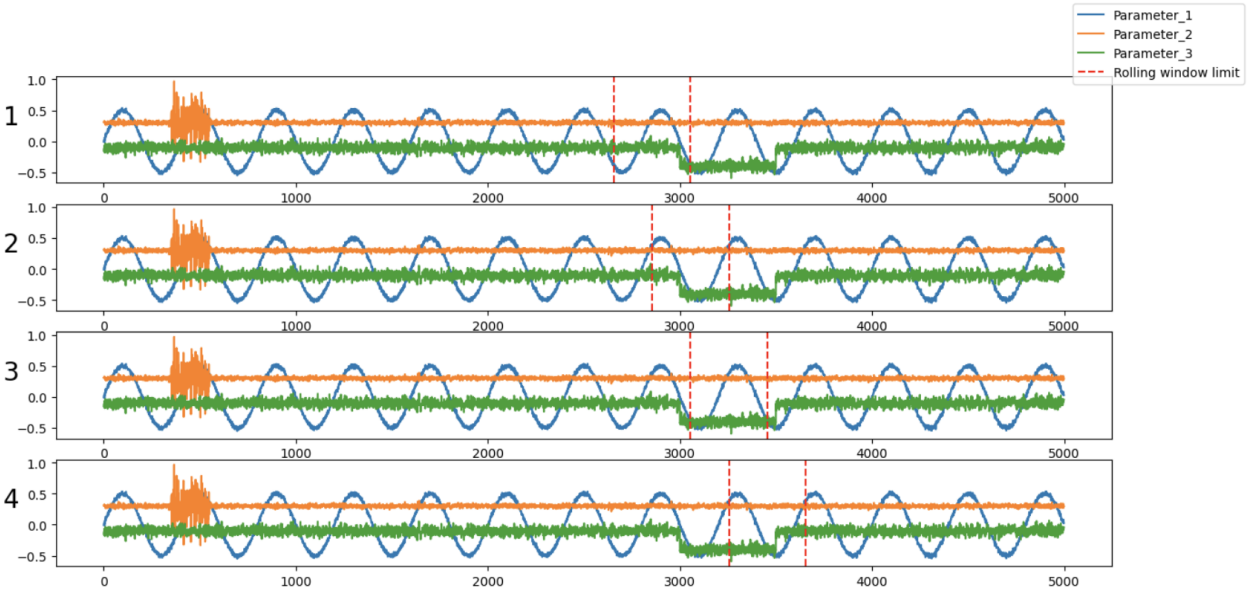


Table 4. Results of *connected components clustering* on data from Figure 5. n_{clust} states how many components are identified initially but before the evaluation we select the final label for each anomalous period as a majority vote. Performance is measured with weighted F_1 metric on merged clusters. In this case, the number of merged clusters corresponds to the true number of clusters (i.e. number of different anomalies in Figure 5). 51 stratified features are selected using Laplacian score.

Feature selection	All features		Laplacian	
Eval. metric	n_{clust}	F_1^W	n_{clust}	F_1^W
TS 1	3	1	4	1
TS 2	4	0.46	4	1
TS 3	2	0.7	6	1
TS 4	4	0.53	4	1
TS 5	3	1	3	1

4. Conclusion

This paper has presented an unusual application of the graph theory technique for clustering purposes in the context of time series multiclass anomaly detection. It helps to identify and group together similar observations when the number of clusters is not known in advance.

From the practical point of view, the main drawback is that a new observation requires to re-compute the connected components which slows down the procedure. Another drawback is that a single isolated observation is necessarily connected to its nearest neighbour.

Despite of these disadvantages, the capacity of the approach to determine number of clusters without apriori information about dataset remains an important benefit for industrial data analysis.

References

[agg] Python skicit-learn: Agglomerative clustering. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html. Accessed: 2023-09-06; default settings.

[Bar02] Albert-László Barabási. Linked: The New Science of Networks. Perseus Books Group, 2002.

[Bar16] Albert-László Barabási. Network Science. Cambridge University Press, 2016.

[BGLL08] Vincent Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 10, 4 2008.

[CFLM22] Mathieu Cura, Katarina Firdova, Céline Labart, and Arthur Martel. Explainable multiclass anomaly detection on functional data. https://arxiv.org/abs/2205.02935, 2022.

[CH74] Tadeusz Calinski and Joachim Harabasz. A dendrite method for cluster analysis. Communications in Statistics, 3(1):1–27, 1974.

[CKLF17] Maximilian Christ, Andreas W. Kempa-Liehr, and Michael Feindt. Distributed and parallel time series feature extraction for industrial big data applications. https://arxiv.org/abs/1610.07717, 2017.

[CMS13] Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. Density-based clustering based on hierarchical density estimates. In Advances in Knowledge Discovery and Data Mining, pages 160–172, 2013.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jorg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, page 226–231, 1996.

[For10] Santo Fortunato. Community detection in graphs. Physics Reports, 486:75–174, 02 2010.

[HCN05] Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. In NIPS'05: Proceedings of the 18th International Conference on Neural Information Processing Systems, 2005.

[MCCH19] Mohammad Malekzadeh, Richard G. Clegg, Andrea Cavallaro, and Hamed Haddadi. Mobile sensor data anonymization. In Proceedings of the International Conference on Internet of Things Design and Implementation, pages 49–58, 2019.

[pyta] Python skicit-learn: Accuracy score. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html. Accessed: 2023-11-10.

[pytb] Python skicit-learn: Normalised mutual information. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.normalized_mutual_info_score.html. Accessed: 2023-11-10.

[Rou87] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20:53–65, 1987.

[Ste57] Hugo Steinhaus. Sur la division des corps matériels en parties. Bulletin of the Polish Academy of Sciences, 4:801–804, 1957.

[Wil96] Robin Wilson. Introduction to Graph Theory, Fourth Edition. Addison Wesley, 1996.

Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, LAMA, 73000 Chambéry, France

Email address: katarina.firdova@univ-smb.fr

Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, LAMA, 73000 Chambéry, France

Email address: celine.labart@univ-smb.fr

Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, LAMA, 73000 Chambéry, France

Email address: laurent.vuillon@univ-smb.fr