IFC-Bank of Italy Workshop on "Machine learning in central banking"

19-22 October 2021, Rome / virtual event

Time series outlier detection, a data-driven approach[1]

Nicola Benatti, European Central Bank,
and Alexis Maurin, Bank of England

[1] This presentation was prepared for the Workshop. The views expressed are those of the authors and do not necessarily reflect the views of the Bank of Italy, the BIS, the IFC or the central banks and other institutions represented at the event.

Time series outlier detection, a data-driven approach

Alexis Maurin, Nicola Benatti

The COVID-19 pandemic has severely impacted the world economy, leading to abrupt changes in collected statistics. It raised the need for new appropriate methodologies to ensure the production of accurate indicators. In this paper, we propose a methodology of macro-economic time series outlier detection, robust to breaks and spikes. By applying unsupervised machine learning techniques, we explore novel ways of identifying abnormal observations in the broad sense, focusing on the dynamic of time series. Clustering algorithms, which aim to group series with similar dynamics, can reveal exogenous information and help us to better detect outliers to be investigated.

Keywords: outlier detection, unsupervised learning, clustering, dynamics finding

JEL classification: C14, C38, C82

Contents

1. Introduction

Motivation and needs

Macro-economic indicators are widely used by researchers and economists on a plethora of topics (e.g. financial accounts, consumer price index, business demographic, labour market) and at different frequencies (monthly, quarterly and yearly). These data are subject to large and unexpected shocks (e.g. economic crises, political or social reforms) which can cause abrupt movements from one period to another. The current COVID-19 pandemic has heavily impacted them and acted as stressor to the classical data quality monitoring procedures such as outlier detection, which would flag considerably more outlying data points than usual.

The classical methodologies to detect outlying data points often use a specified threshold on the growth rate, or forecast a value (e.g. ARIMA models) and verify whether the real value lies within the estimated confidence interval or not. However, it is quite intricate to define the proper threshold on growth rates above which a data point should be defined as an outlier, while the forecast-based approach can only be applied to the most recent data points and does not use the information from the latest data points. Beyond that, these methods are solely considering a univariate approach. They are not using the possible relations between the series, which can add valuable information with regards to the identification of outliers. There exist advanced time series forecasting algorithms which include the modelling of exogenous variables (ARIMAX model) or operate in a multivariate manner (VAR/VARMA models). The VARMAX model even combines the two approaches. These techniques can be efficient for specific cases but they require an advanced data treatment, thus restraining their generalised implementation, especially for automated procedures.

We explored data-driven ways to create an outlier detection procedure, robust to systemic breaks and spikes, applicable to any macro-economic time series data, with the aim of better detecting series with abnormal behaviour which might either come from reporting error or strong temporary factors.

Testing of classical methodologies

In this paper, we focus our interest on employment data from the National Accounts, which is described in Section 2.

Growth rate

Figure 1 displays two annual series from Spain: the left graph depicts the evolution of self-employed jobs in the G[1], H[2] and I[3] sectors and the right one depicts the evolution of the number of employees (as persons) in the F[4] sector. The complete description

[1] Wholesale and retail trade; repair of motor vehicles and motorcycles

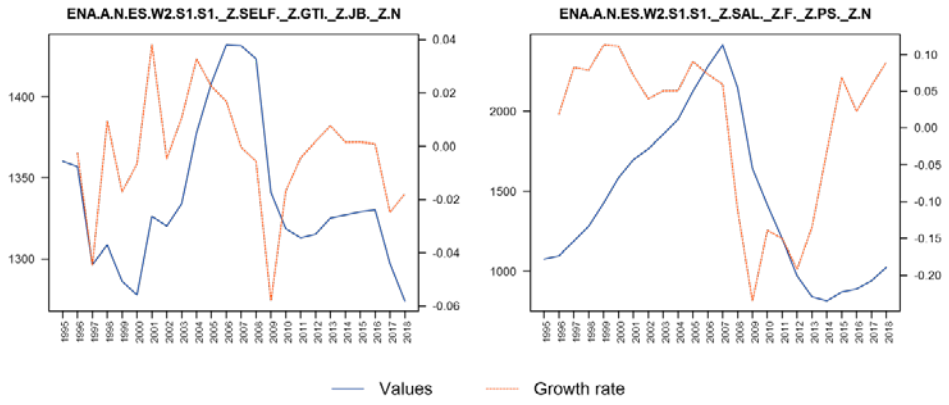
[2] Accommodation and food service activities

[3] Transportation and storage

[4] Construction

list of sectors defined following the NACE rev. 2 classification is available in Annex A. For illustration purposes, we processed annual data from 1995 to 2018. The values plotted as a blue solid line are reflected on the left y-axis. The growth rates plotted as an orange dashed line are reflected on the right y-axis.

Figure 1. Examples of annual series with abrupt decrease



The order of magnitude of the growth rates can differ between series: from -0.06 to 0.04 for the series on the left-hand side, from -0.22 to 0.10 for the series on the right-hand side. This can result in challenges with regards to the definition of the appropriate threshold above which a certain data point should be defined as an outlier. We need more information (e.g. what the macro-economic environment in this country for this sector is) to correctly identify an unusual change, and thus cannot rely on a univariate method.

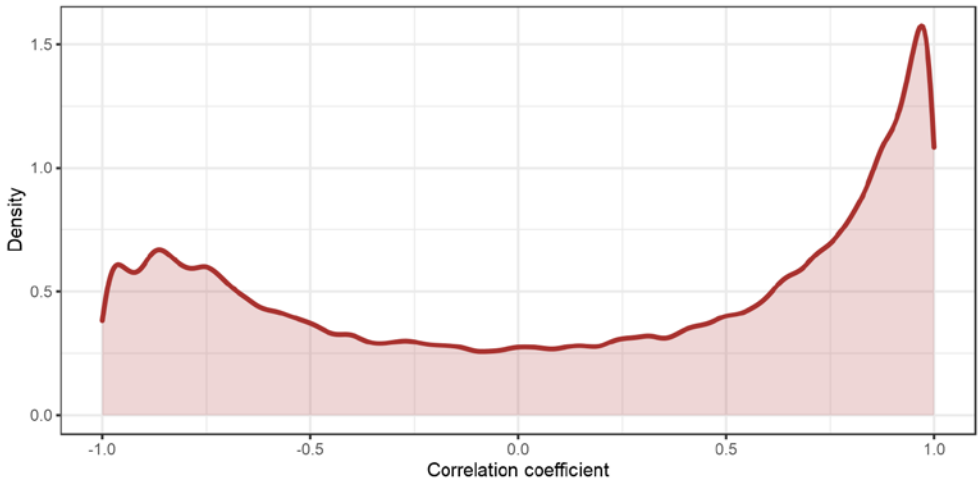
Time series modelling

We fitted ARIMA models on quarterly data, from 1995Q1 to 2020Q1, and forecasted the value for 2020Q2, a quarter that was heavily impacted by the COVID-19 pandemic. We then checked if the real value lies in the estimated confidence interval which was set to 99.5%, to relax the classical outlier rule. Among the 6638 series, 2957 were flagged as series containing outlying data points for 2020Q2. In comparison, when fitting ARIMA models from 1995Q1 to 2019Q3 and forecasting 2019Q4, only 149 series were flagged. The ARIMA models behave quite well in general but are overwhelmed in cases where external shocks impact and drive the series' movement, therefore requiring a procedure which can overcome this situation.

Our approach

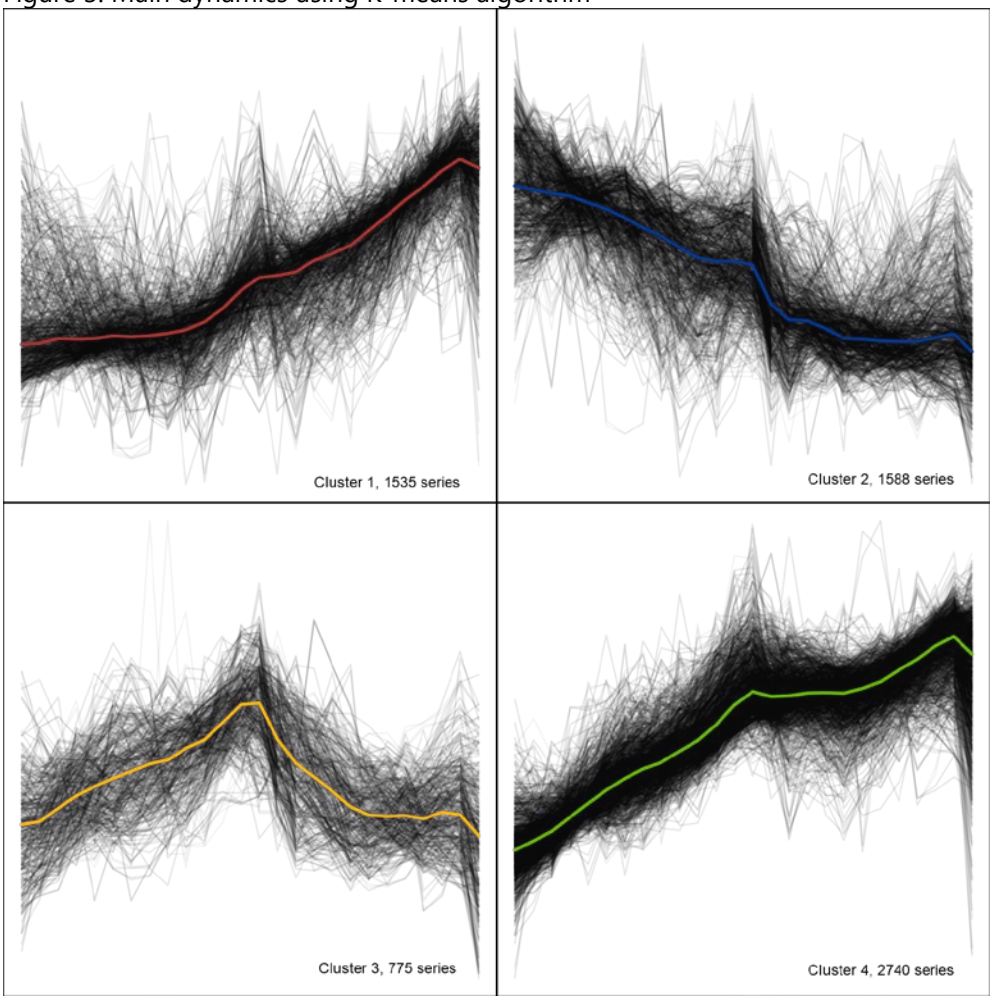
Macro-economic datasets cover topics broken down into multiple dimensions and therefore often exhibit relations between the time series that compose each dataset. By using the Spearman correlation, we can measure the rank correlation and observe monotonicity between the series. Figure 2 plots the density function of the correlation coefficients across the quarterly employment series of the National Accounts, from 1995Q1 to 2021Q1, smoothed with the LOWESS algorithm described in Section 3. We can observe higher masses for strongly correlated series, implying strong monotonic relationships. The lower mass around 0 indicates low correlation, i.e. of the uniqueness of one series' dynamic. We have a first evidence of groupable series.

Figure 2. Spearman correlation density curve



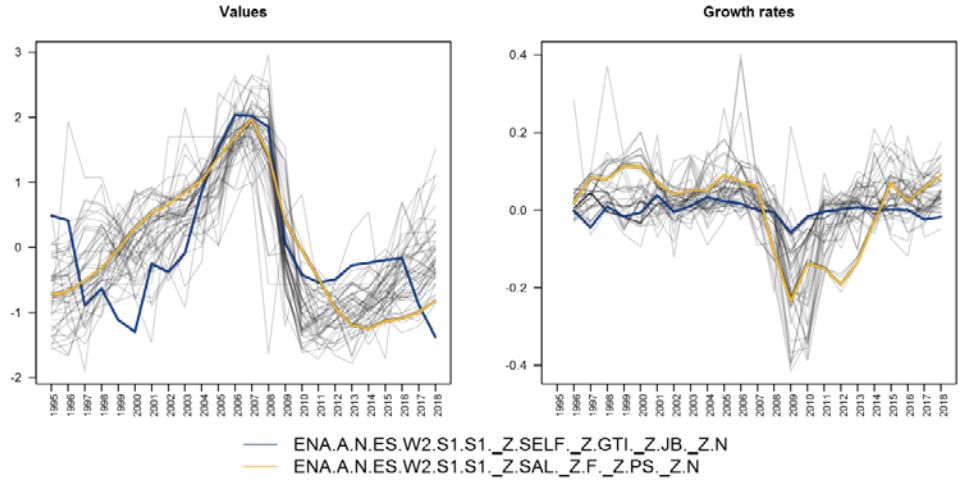
Going further into this inspection, we can identify the main dynamics. The K-means algorithm is one of the simplest and most popular clustering algorithms which enables the grouping of similar observations, series in our case. Figure 3 plots the four main patterns found in the annual data from 1995 to 2020, extracted from the quarterly data, which corroborate the presence of dynamics (median series coloured).

Figure 3. Main dynamics using K-means algorithm



The evidence of relationships between the time series can be used to answer questions the classical methodologies cannot easily solve: "Is a large movement an actual outlier or a response to an externality?". Basing our decision-making on movements for each series requires advanced expertise and would involve a tremendous amount of time as seen with the two series shown previously. Therefore, to overcome this issue in a generalised fashion, our approach is to cluster series with similar dynamics. The series from Figure 1 lie in the same cluster, hence exhibiting similar dynamics despite different values and growth rates. Figure 4 plots all series which lie within this cluster. We can see that, by focusing on the scaled series (i.e. dynamics) instead of inspecting the large growth rates' range, we can target the series with an abnormal behaviour better. For these specific series, the Great Financial Crisis is causing the drop in growth rates in 2009.

Figure 4. Example of a cluster of series with similar dynamics



The left graph displays the scaled series. The right graph depicts their respective growth rates. The series with an absolute growth rate greater than 0.5 (50%) have been removed for visibility. The cluster comes from the procedure described in Section 4.

Objectives

The objective is to create an outlier detection tool applied to macro-economic data, as automated as possible, in order to help narrow down data points that would be identified as outliers and would consequently require further investigation. The approach is completely data-driven, in other words, we let the data speak for itself. The initial and only assumption made is that the series can be grouped with respect to their dynamic. We only use historical data and do not use any additional information with regards to the type of outlier we are seeking prior to the application of the methodology. We try to mimic the human experience in the identification process and rather nowcast than forecast. Subsequently, we look for the data points that differ the most from the cluster to which they have been allocated. They might present some peculiarity as they are not where they are expected to be.

The data is presented in Section 2, the statistical tools used are defined in Section 3, the procedure is described in Section 4 and the results are reported in Section 5.

2. Data

The National Accounts are compiled according to the accounting definitions and methodologies set out in the ESA 2010 Regulation. Each quarter, Eurostat publishes several macro-economic aggregates, including employment data with industry breakdowns, what we call Employment National Accounts (ENA). The ENA data covers 37 countries[5] on annual and quarterly frequencies, broken down into three dimensions:

- Branches of economic activities (NACE rev. 2 classification, description available in Annex A)
- Employment status (employees and self-employed) and totals (total employment and total population)
- Unit of measure (jobs, persons, hours worked and full-time equivalent)

Due to differences in data availability between countries, all dimensions described above are not evenly represented across countries. We focused on quarterly data from 2011Q1 to 2021Q1, and did not use European aggregates. In total, we included 6638 series covering 31 countries.

3. Statistical tools

This project was undertaken with the R software and the different statistical tools used are presented and described below.

Smoothing method – LOWESS

The Locally Weighted Scatterplot Smoothing (LOWESS) is a non-parametric regression tool that fits a smooth curve to data points. It is an iterative process which fits local polynomials on a sliding window using weighted least squares (WLS).

First, it applies weights with respect to the abscissa closeness, giving more weight (or influence) to the points that are closest to the estimated one. Second, it adjusts these fitted values based on their distance from the actual ones, adding additional weights to the WLS. This second step can be repeated multiple times, until the curve is sufficiently smoothed.

The LOWESS algorithm requires setting different parameters: the polynomial degree, a weight function, the number of iterations and the window size (i.e. smoothing parameter). The smoothing parameter defines the proportion of data points to be used within the windows in order to fit each polynomial. Larger values lead to more smoothness.

[5] AL, AT, BE, BG, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LI, LT, LU, LV, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, TR

We use the lowess() function from the stats library which fits polynomials of degree 1 (i.e. lines) and defines weights with Tukey's biweight function (with a cut-off set at 6 times the median absolute deviation of the residuals). The default number of iterations is set to 3. We use this tool to smooth the time series prior to the clustering algorithm with the intention of reducing the noise (e.g. removing seasonality) and catching only their main dynamic, as this methodology gives a robust estimation of the outliers through the weighting and iteration process.

Metrics

The clustering algorithms require a distance metric in order to compute the dissimilarities between the observations and thereby allowing to group the similar ones together. Contingent upon the type of data we are using, we have selected two distances: the Minkowski distance for cases with only quantitative variables and the Gower distance for cases with both quantitative and qualitative variables.

Minkowski distance

In cases for which only quantitative data needs to be processed, we apply the Minkowski distance which is defined as follows:

$$D_{Minkowski}(x_i, x_j, p) = \left(\sum_{v=1}^{V} |x_i^v - x_j^v|^p \right)^{\frac{1}{p}}, for\ V\ variables.$$

For our methodology, we apply this distance with p=2 (Euclidean distance). We use the dist() function from the stats library to compute this distance.

Gower distance

In cases for which both quantitative and qualitative variables compose our dataset, we have to use a different metric that can handle these two types of data simultaneously: the Gower distance. It is defined as follows:

$$D_{Gower}(x_i, x_j) = \frac{1}{V} \sum_{v=1}^{V} d_{ij}^v, for\ V\ variables,$$

where $d_{ij}^v \in [0,1]$ is the partial dissimilarity between observations i and j for the variable v.

Depending on the type of the variable (qualitative or quantitative), the partial dissimilarity is computed differently. For the quantitative variables, it is defined as the ratio of the absolute difference and the maximum range observed:

$$d_{ij}^v = \frac{|x_i^v - x_j^v|}{\left|\max(x^v) - \min(x^v)\right|}$$

For the qualitative variables, the dissimilarity takes the value 0 if the observations are the same and 1 otherwise:

$$d_{ij}^{v} = \begin{cases} 0 \; if \; x_i^v = x_j^v \\ 1 \; otherwise \end{cases}$$

We use the daisy() function from the cluster library to compute this distance.

Pre-processing

To avoid the scaling effect between features, we need to standardise our data. The Gower distance does not need a preliminary transformation as it is already integrated in the formula with the min-max scaling at the denominator. However, the Minkowski distance needs to be processed. We use the Z-score normalisation:

$$x_i^{k'} = \frac{x_i^k - \mu^k}{\sigma^k}$$

where μ^k is the mean value of the feature k and σ^k the standard deviation of the feature k.

Clustering algorithms

Clustering is an unsupervised machine learning technique with the goal of grouping observations with similar characteristics. We use two clustering algorithms in accordance with the objective we are trying to reach:

- The Affinity Propagation algorithm to identify the main dynamics of the series;
- The DBSCAN algorithm to detect the outlying data points.

Affinity Propagation

The Affinity Propagation algorithm, proposed by Frey and Dueck in 2007, is a clustering method based on the concept of "message passing". The algorithm has a graph-based approach, in other words, it considers each observation as nodes of a network between which “messages” are being exchanged. For each observation, the goal is to find the one that is the most representative of its cluster: its exemplar. Let us note that one observation’s exemplar can be itself. To exchange these "messages", the algorithm uses 3 matrices:

- The Similarity matrix $s(i,j)$ which measures the similarity between the observation i and j. We use the negative squared Euclidean distance:

$$s(i,j) = -\left\|x_i - x_j\right\|^2$$

- The Responsibility matrix $r(i,j)$ which quantifies the extent to which the observation j is suited to be the exemplar of the observation i, taking into account other potential exemplars.
- The Availability matrix $a(i,j)$ which quantifies the extent to which the observation i should choose j as its exemplar, taking into account other potential exemplars.

The algorithm is initialised by considering all the data points as a potential exemplar and takes two main parameters as input: the preference vector and the damping factor. The number of clusters does not need to be specified which yield the Affinity Propagation more favourable to the other classical algorithms.

The preferences vector $s(i,i)$ represents the a priori suitability of a data point to be an exemplar. Its value directly influences the number of clusters and a high value per observation will increase the propensity to be chosen as an exemplar. In case of no a priori knowledge, the vector can be set as any quantile of input similarities, typically the median.

During the message-passing procedure, numerical oscillations may occur in some cases, leading the algorithm not to converge. The damping factor (λ) comes into play. At each iteration t, the messages are damped by this factor as follows:

$$r_t(i,j) = \lambda \cdot r_{t-1}(i,j) + (1-\lambda) \cdot r_t(i,j)$$

$$a_t(i,j) = \lambda \cdot a_{t-1}(i,j) + (1-\lambda) \cdot a_t(i,j)$$

λ takes values between 0 and 1.

After having defined these parameters, the algorithm procedure can begin:

1. Set the responsibility and availability matrices to 0

2. Repeat until convergence (number of iterations reached or exemplar remains unchanged for a defined number of iterations):

   - Update responsibility

$$r(i,k) \leftarrow s(i,k) - \max_{k' \neq k}\{a(i,k') + s(i,k')\}$$

   - Update availability

$$a(i,k) \leftarrow \min_{i' \neq k}\left\{0, r(k,k) + \sum_{i' \notin \{i,k\}} \max\{0, r(i',k)\}\right\} \; for \; i \neq k$$

$$a(k,k) \leftarrow \sum_{i' \neq k} \max\{0, r(i',k)\}$$

3. Extract the exemplar of each observation to get the clusters:

$$examplar_i = \underset{k}{\operatorname{argmax}}\{a(i,k) + r(i,k)\}$$

We use the apcluster() function from the apcluster library.

DBSCAN

The DBSCAN (Density-Based Spatial Clustering and Application with Noise), proposed by Ester, Kriegel, Sander and Xu in 1996, is a density-based clustering algorithm. Its goal is to seek high density areas, which are then defined as clusters. Similar to the Affinity Propagation algorithm, the number of clusters does not need to be specified and has two main parameters to set:

- The epsilon (eps) defines the radius around an observation in which other observations will be defined as neighbours.
- The minimum points (MinPts) defines the minimum number of neighbours required within an observation's radius to form a dense region.

A distance metric must be defined to measure the closeness between the observations, then the algorithm works as follows:

- For each observation, it computes its distance to all the other observations and counts how many are falling into the epsilon radius. If the count is inferior to the MinPts parameter (but not zero), the observation is marked as a border point. If the count is superior or equal, it is marked as a core point. Finally, if it has no neighbour, it is defined as noise.
- If a core point is not assigned to a cluster, a new one is created. Through a chaining process, all the connected core points are found and assigned to this cluster.
- Finally, it allocates each border point to the closest connected cluster.

We used the dbscan() function from the dbscan library.

4. Procedure

Dynamics finding

We first use a clustering algorithm to catch the unobservable information represented by the different dynamics. The main challenge here is to define the optimal number of patterns (i.e. clusters) as the partitioning algorithms require to set the number of clusters as a parameter and the hierarchical algorithms require to define a threshold from the dendrogram. Our need for an automated procedure led us to look for other types of clustering approaches as the previously cited ones were not efficient nor easily optimisable. We compared several algorithms[6] and based on metrics (GAP value, silhouette coefficient and Dunn Index) and pertinence of the clusters, we decided to opt for the Affinity Propagation algorithm which had demonstrated simplicity, applicability (i.e. automation) and performance.

Before running the clustering algorithm, we normalise the data and apply the LOWESS algorithm with a window size of 20%. This smoothing aims to get rid of the short-term variations and only grasps the general dynamic over the whole time span.

In order to get the minimum number of clusters, the "preference" parameter is set to the minimum (0), giving the minimum distance between points to the preference vector. We set the damping factor λ to 0.5, to control oscillations and ensure convergence of the algorithm.

[6] K-means, K-medoid, Fuzzy C-means, HDBSCAN, DBSCAN, OPTICS, MeanShift, SOM, EM

Outliers identification

The rationale behind our approach is the following: within each cluster found by the Affinity Propagation algorithm, we apply the DBSCAN algorithm over all series, and we expect that for each period, the data points should lie in the same density area. Therefore, any inconsistency in the allocation of a data point will set it as outlying. It also allows us to track down outliers over the entire series, or to focus on a certain segment (e.g. latest data points).

To do so, for each cluster, we define as variables the scaled value of a data point for a given period and the period in which it falls. By computing the Gower distances on these two variables, two data points lying in the same period will have a distance between 0 and 0.5 (as the distance for the period will be equal to 0) and two data points in different periods will have a distance between 0.5 and 1. Therefore, a data point far from the main density area, where the data points of the same period lie, will be defined as an outlier.

We look for the most outlying data points for each cluster and designed the above operation as follows:

1. Convert wide to long format, adding the period
2. Compute the Gower distance between each data point.
   In case the number of data points is too high, a sliding window is set and the distances are computed over the defined portion.
3. Run a DBSCAN algorithm with parameters set as:
   - eps = largest minimum distance between two points
   - MinPts = 1
4. Get the number of outlying points
5. While no data points are spotted as outliers
   - Set: $eps = eps * 0.95$
   - Re-run a DBSCAN
6. We proceed up to 5 iterations

5. Results

We focused on the data of the last 10 years, from 20211Q1 to 2021Q1, as we only wanted to depict the short/medium term dynamics of the series. The Affinity Propagation algorithm identified 39 clusters. They are displayed in Annex B. Within each cluster, we ran the outlier identification from 2019Q4 to 2021Q1 and found 126 outlying data points. They are displayed in Annex C. For comparison, we fitted ARIMA models and flagged outliers using the 95% confidence interval rule. A total of 10716

data points were flagged. Our procedure shows a more efficient way to reduce the time an expert will investigate into these outliers at a granular level.

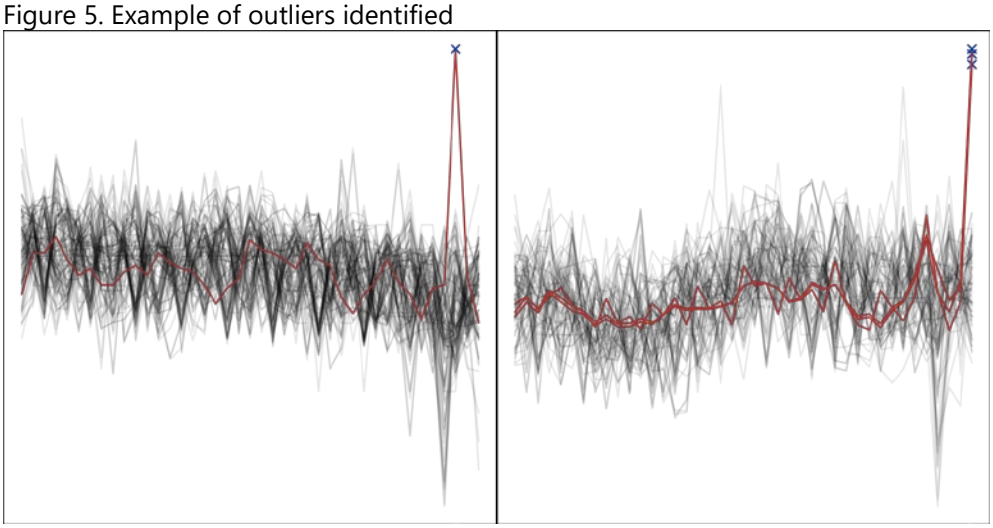
Table 1. Number of outliers found per period

	2019Q4	2020Q1	2020Q2	2020Q3	2020Q4	2021Q1	Computation time
ARIMA model	430	1563	3356	2557	1241	1369	5 hours 30 mins
Our procedure	7	7	52	6	19	35	13 mins

Examples of outliers

Figure 5 plots two examples of clusters in which the outliers flagged are showing a peculiar pattern compared to the cluster they lie in.

Figure 5. Example of outliers identified

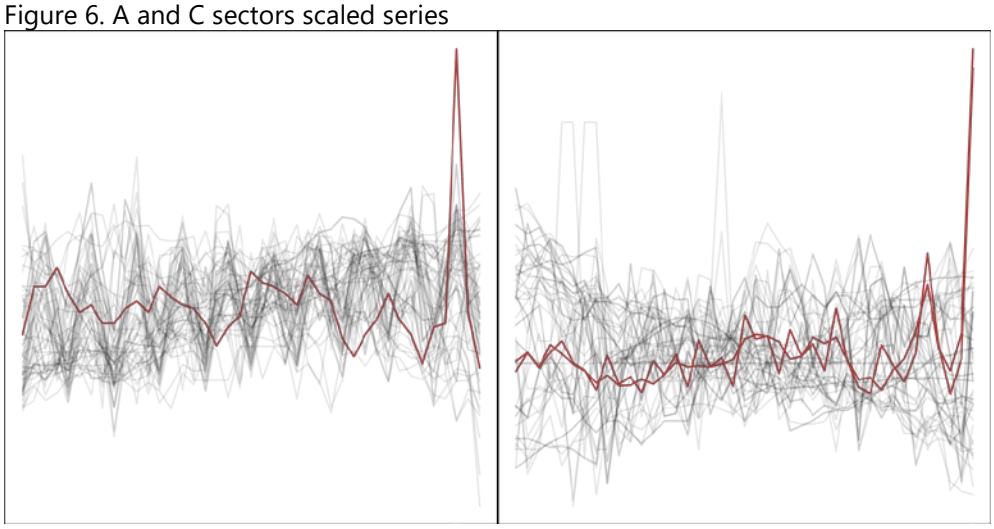


The left cluster groups series which exhibit a slowly decreasing trend over time. While some series are showing an important decrease in 2020Q2, they recover quickly to the pre-shock level. One outlier is flagged in 2020Q3 as it is the only one displaying an abrupt increase. The outlier comes from the series depicting the number of employed persons in the A sector in a specific country. The right cluster groups series which exhibit a slow decrease from 2011Q1 to 2014Q1, followed by an increase until 2015Q2 and a stable trend until 2021Q1. While some series are showing a significant decrease in 2020Q2, they recover quickly to the pre-shock level. Three outliers are flagged in 2021Q1 as they are the only ones displaying a significant increase. The outliers come from the series depicting the total number of self-employed persons in the C sector (both seasonally adjusted and non-adjusted) as well as in the B, C, D and E sectors combined in a specific country.

We can further examine these series by comparing them to other series compiling the same statistic but from the other countries. Figure 6 plots sector A series on the left and the series of sector C on the right, with the outliers coloured. For sector A, only the spotted series shows the increase in 2020Q3. The rationale behind this increase for this country might be the compensation for the temporary suspension of fish activities to support commercial fishers who have been affected by the COVID-19 pandemic, coupled with the temporary easing of regulations covering aquaculture

licences. For sector C, only the spotted series show the increase in 2021Q1. The increase for this country might be driven by the ongoing reopening of the economies and continued strength in demand from both domestic and European countries, combined with measures in response to COVID-19. The main measures to preserve employment and support household income for this country are special short-time work schemes and the extraordinary allowance for the self-employed (e.g. moratorium on tax debt and social security contributions, the deferral of tax payments).

Figure 6. A and C sectors scaled series



Our procedure solely uses historical data so that it can be run with any macro-economic data. On top of that, it allows to be less restrictive when it comes to the identification of outliers and rather flag too many data points than missing potential outliers. The complementary investigation carried out above could be implemented to enhance the outlier identification for our dataset, as well as the incorporation of additional information into the procedure (e.g. sectoral information).

6. Applications

The model we proposed has the characteristics of being completely data-driven, of considering long term dynamics of correlated series and, more important, of allowing for a break in the linearity of these correlations if this happens for a specific period as it was the case at the beginning of the COVID-19 pandemic. Given its wide range of applications on any type of time-series, the model can be considered to work independently or to be integrated into other tools. For example, in order to obtain more detailed explanations of why one observation is considered to be an outlier, we apply the feature-additive ranking technique on the observations spotted as outliers. This technique consists in running a XGBoost model using the outlying series spotted by the model presented above as dependent variable and using the remaining series in our dataset as features. A HDBSCAN model is then run on the residuals of the estimation to confirm outliers, Shapley values are calculated and aggregated to explain the model. The use of the time series clustering approach proposed in this paper allows us to filter a-priori the series to use as dependent variable and therefore maintain the process as computationally lightweight as possible.

7. Conclusion

This paper presents an approach for outlier detection for macro-economic datasets using unsupervised machine learning, robust to external shocks. Our procedure enables the reduction of the list of potential outliers, the ones with dynamics that differ the most from the general trends. It shows more efficiency compared to classical methodologies, especially amidst unstable periods such as the current COVID-19 pandemic. It can also be used as an investigation tool to identify time series with unusual movements in the broad sense, rather than looking for potential errors. The source code is available at *https://github.com/alexismaurin/ODMS*.

This approach is very generic as it only uses historical data and makes it applicable to any macro-economic data. The clustering process as well as the outlier identification can be adapted with respect to the data processed. This can be enhanced by adding exogenous data, in order to get more representative clusters and better target the data points that should be flagged as outliers.

Annexes

Annex A

Countries abbreviations

BE	Belgium	**HR**	Croatia	**PL**	Poland
BG	Bulgaria	**IT**	Italy	**PT**	Portugal
CH	Switzerland	**IS**	Iceland	**RO**	Romania
ME	Macedonia	**CY**	Cyprus	**AL**	Albania
CZ	Czech Republic	**LI**	Liechtenstein	**SI**	Slovenia
DK	Denmark	**LV**	Latvia	**SK**	Slovakia
DE	Germany	**LT**	Lithuania	**RS**	Serbia
EE	Estonia	**LU**	Luxembourg	**MK**	North Macedonia
IE	Ireland	**HU**	Hungary	**FI**	Finland
GR	Greece	**MT**	Malta	**SE**	Sweden
ES	Spain	**NL**	Netherlands	**TR**	Turkey
FR	France	**NO**	Norway		
GB	Great Britain	**AT**	Austria		

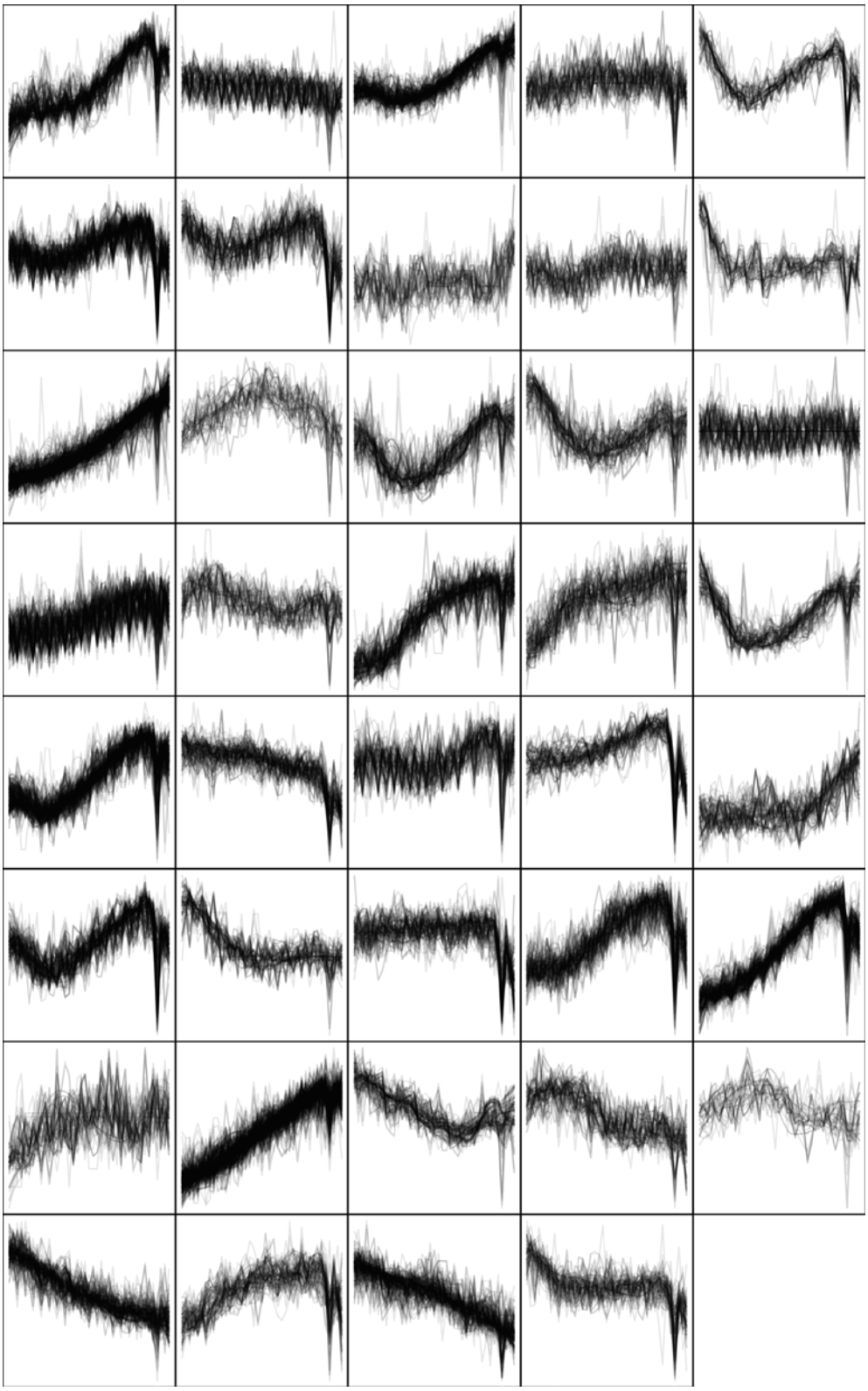
In accordance with EU practice, the EU Member States are listed in this report using the alphabetical order of the country names in the national languages.

NACE rev. 2 classification

Section	Description
A	Agriculture, forestry and fishing
B	Mining and quarrying
C	Manufacturing
D	Electricity, gas, steam and air conditioning supply
E	Water supply, sewerage, waste management and remediation activities
F	Construction
G	Wholesale and retail trade; repair of motor vehicles and motorcycles
H	Accommodation and food service activities
I	Transportation and storage
J	Information and communication
K	Financial and insurance activities
L	Real estate activities
M	Professional, scientific and technical activities
N	Administrative and support service activities
O	Public administration and defence; compulsory social security
P	Education
Q	Human health and social work activities
R	Arts, entertainment and recreation
S	Other service activities
T	Activities of households as employers; undifferentiated goods- and services-producing activities of households for own use
U	Activities of extraterritorial organisations and bodies

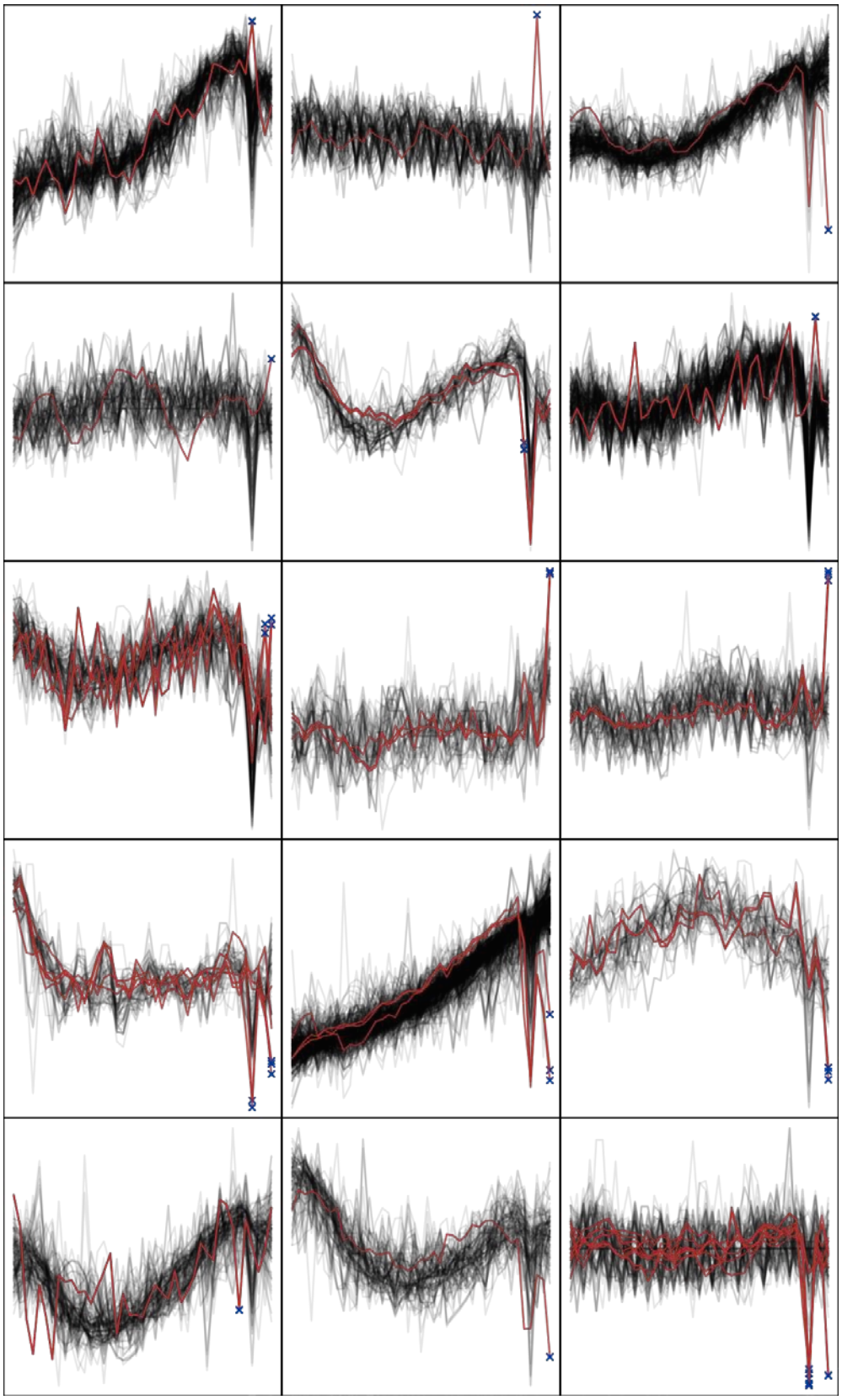
Annex B

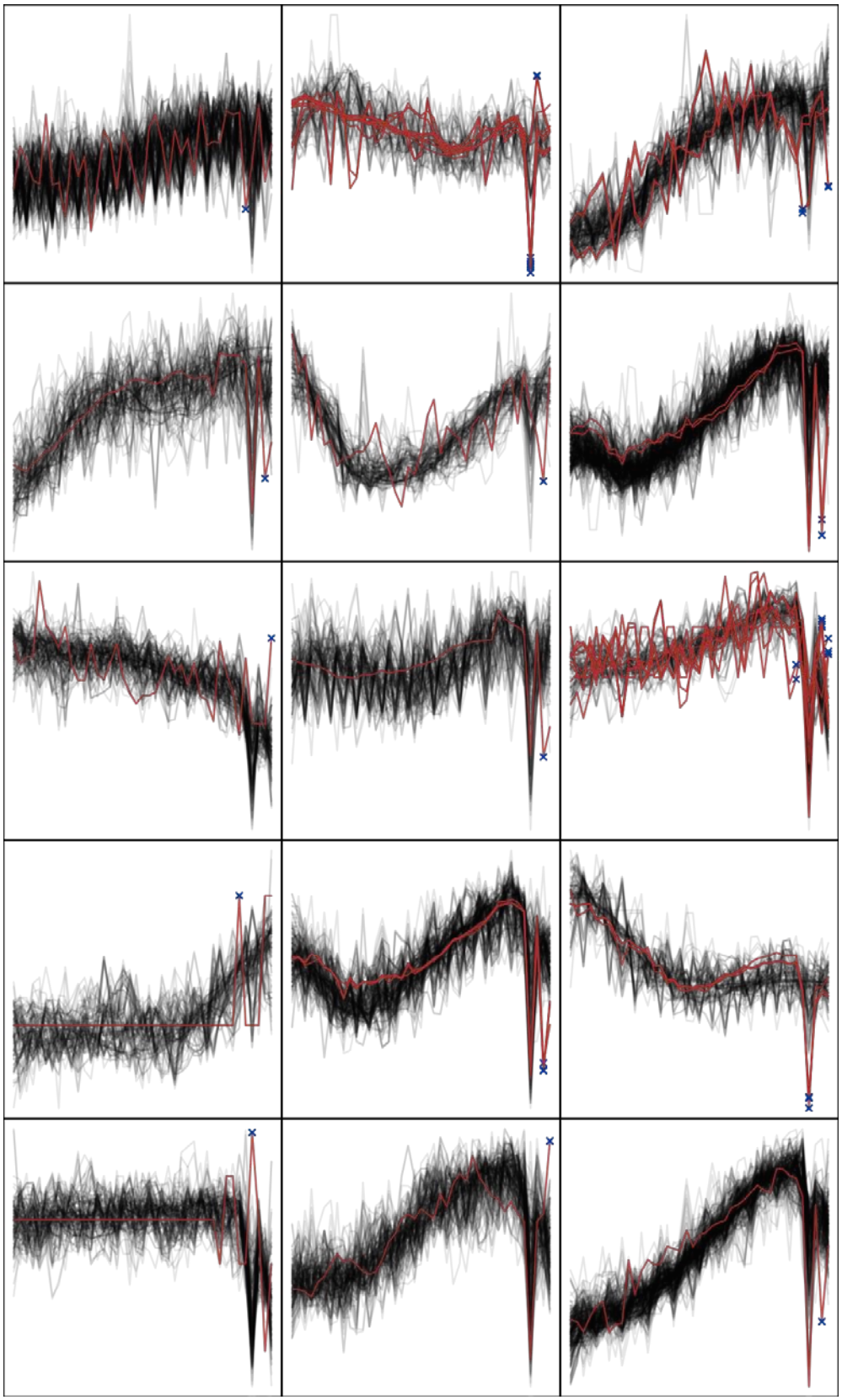
Clusters found by the Affinity Propagation algorithm from 2011Q1 to 2021Q1

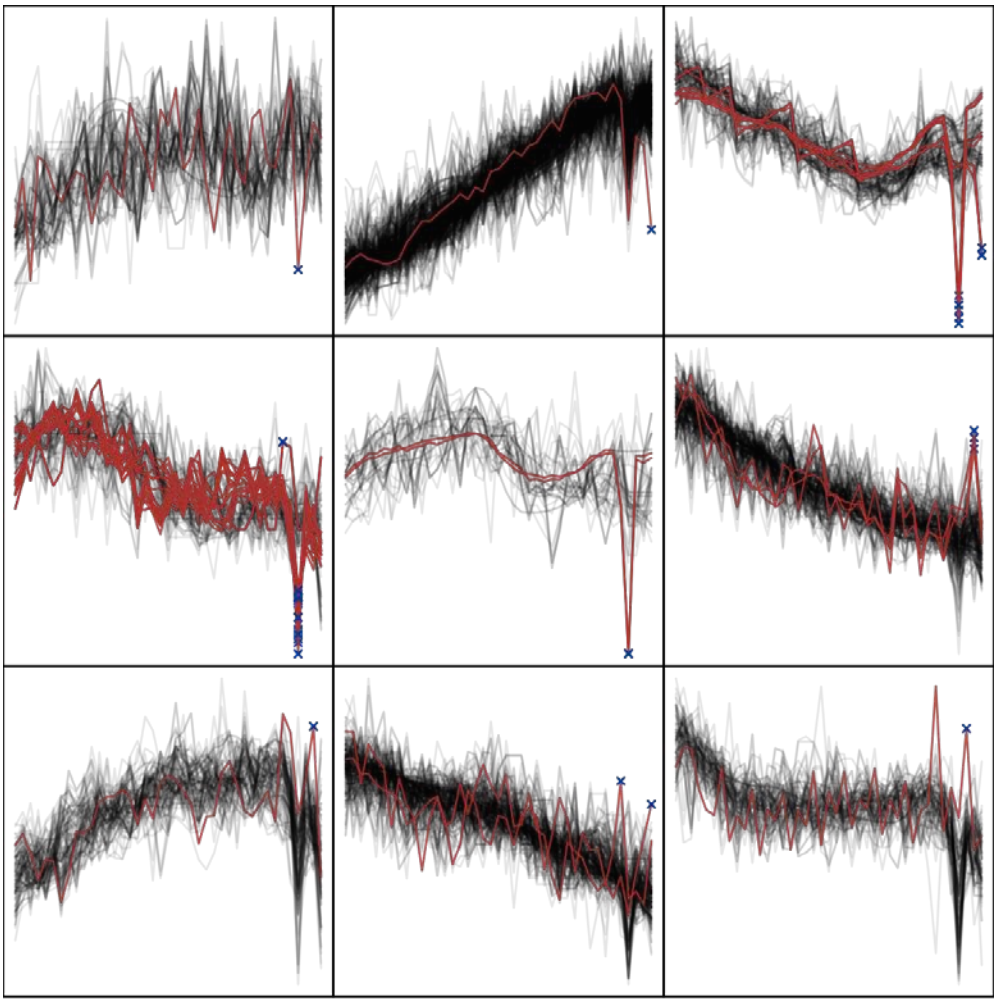


Annex C

Outliers found in each cluster from 2019Q4 to 2021Q1







References

Atkinson A.C., Koopman S.J. and Shepard N. (1997) "Detecting shocks: outliers and breaks in time series", Journal of Econometrics 80, 387-422

Benatti N. (2019) "A machine learning approach to outlier detection and imputation of missing data", IFC Bulletins chapters, Bank for International Settlements (ed.), Are post-crisis statistical initiatives completed?, volume 49, Bank for international Settlements.

Brendan J.F. and Delbert D. (2007) "Clustering by Passing Messages Between Data Points", Science 315, 972-976.

Cleveland W.S. (1979) "Robust locally weighted regression and smoothing scatterplots", Journal of the American Statistical Association 74, 829-836.

Ester M., Kriegel H., Sander J. and Xu X. (1996) "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", Institute for Computer Science, University of Munich. *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, 226-231.

Goin D.E. and Ahern J. (2019) "Identification of Spikes in Time Series" Epidemiologic Methods 8

Gower J.C. (1971) "A general coefficient of similarity and some of its properties", Biometrics 27, 857-874.

Hyndman R.J. and Khandakar Y. (2008) "Automatic time series forecasting: The forecast package for R", Journal of Statistical Software, 26(3).

Kaufman L. and Rousseeuw P.J. (1990) "Finding Groups in Data: An Introduction to Cluster Analysis", Wiley, New York.

MacQueen J. (1967) "Some methods for classification and analysis of multivariate observations, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and* Probability", Berkeley, University of California Press, 1:281-297.

Struyf A., Hubert M. and Rousseeuw P.J. (1997) "Integrating Robust Clustering Techniques in S-PLUS", Computational Statistics and Data Analysis 26, 17-37.

BANK OF ENGLAND

IFC Conference 19th-22nd Oct. 2021

Time series outlier detection, a data-driven approach

The project was carried out when both authors were working at the European Central Bank

Alexis Maurin - Bank of England
Nicola Benatti - European Central Bank

Overview

- Data
- Motivation & Needs
- Approach & Goal
- Procedure
- Results & Application
- Conclusion

Employment National Accounts (ENA)

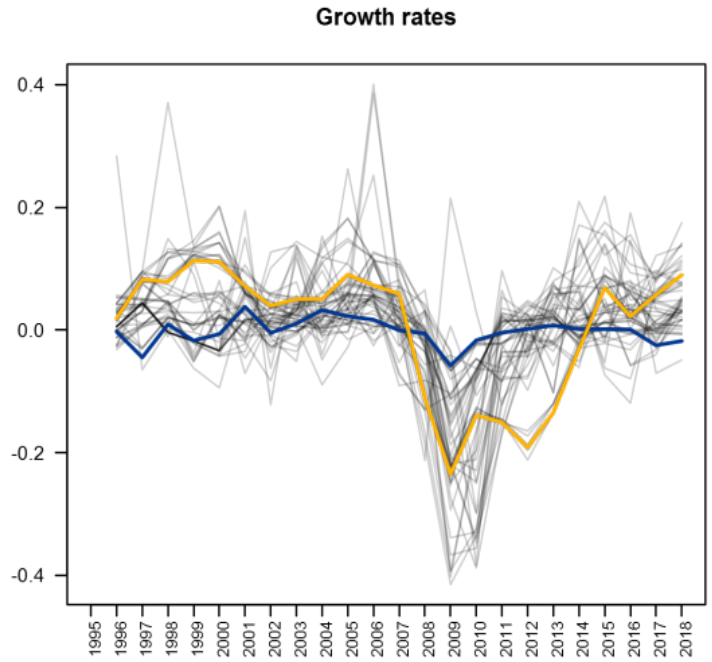
Quarterly data, broken down into three dimensions:

- Branches of economic activities
- Employment status (employees and self-employed) and totals (total employment and total population)
- Unit of measure (jobs, persons, hours worked and full-time equivalent)

- Total of 6638 series covering 31 countries

Motivation and needs

- Macro-economic indicators subject to unexpected shocks
- COVID-19 pandemic heavily impacted their movement
- Data quality monitoring procedures challenged
- Classical Methodologies :
  - Threshold on growth rate:
    Complex to define
  - Univariate Time Series Forecasting:
    20x more outliers flagged during shocks

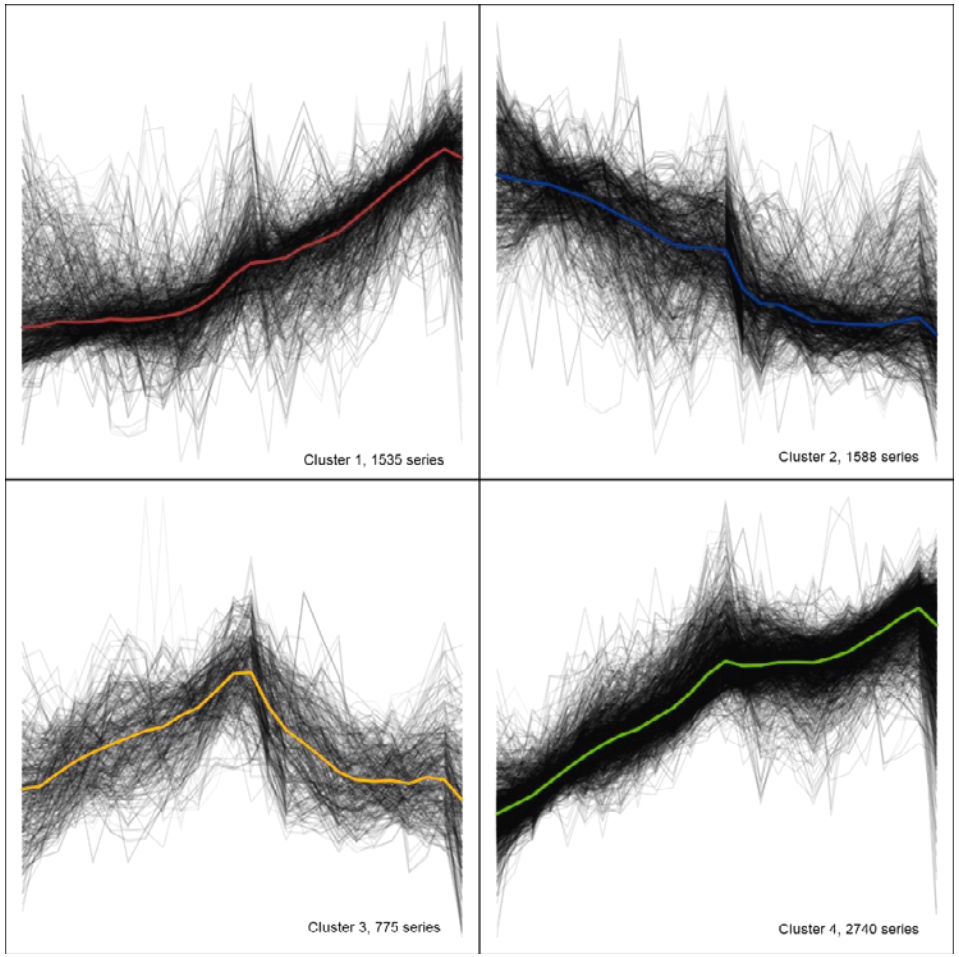


Approach and Goal

- Correlations within dataset
- Identify the main dynamics → **Clustering**
- Detect outliers within each cluster

Goal:

- Robust to systemic spikes and breaks
- Applicable to any macro-economic dataset
- High level automation

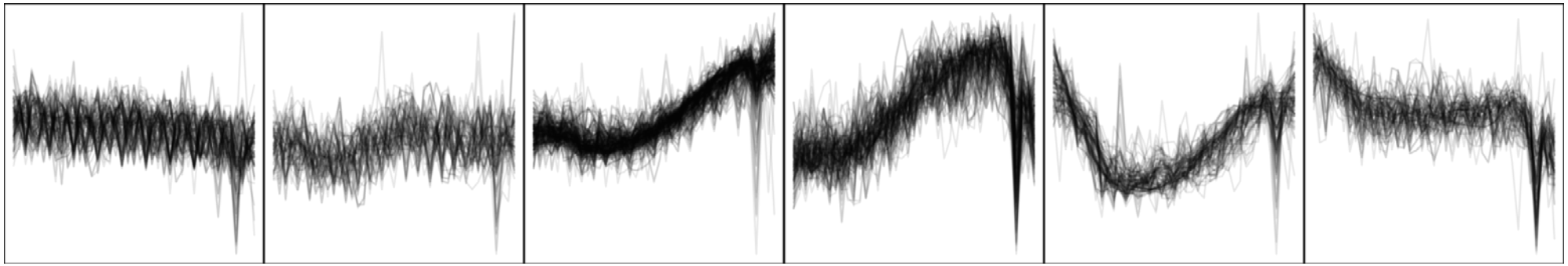


Main dynamics using K-means algorithm

Procedure

1. Standardise the data and smooth the series with the **LOWESS** algorithm

2. Identify clusters (dynamics) with the **Affinity Propagation** algorithm

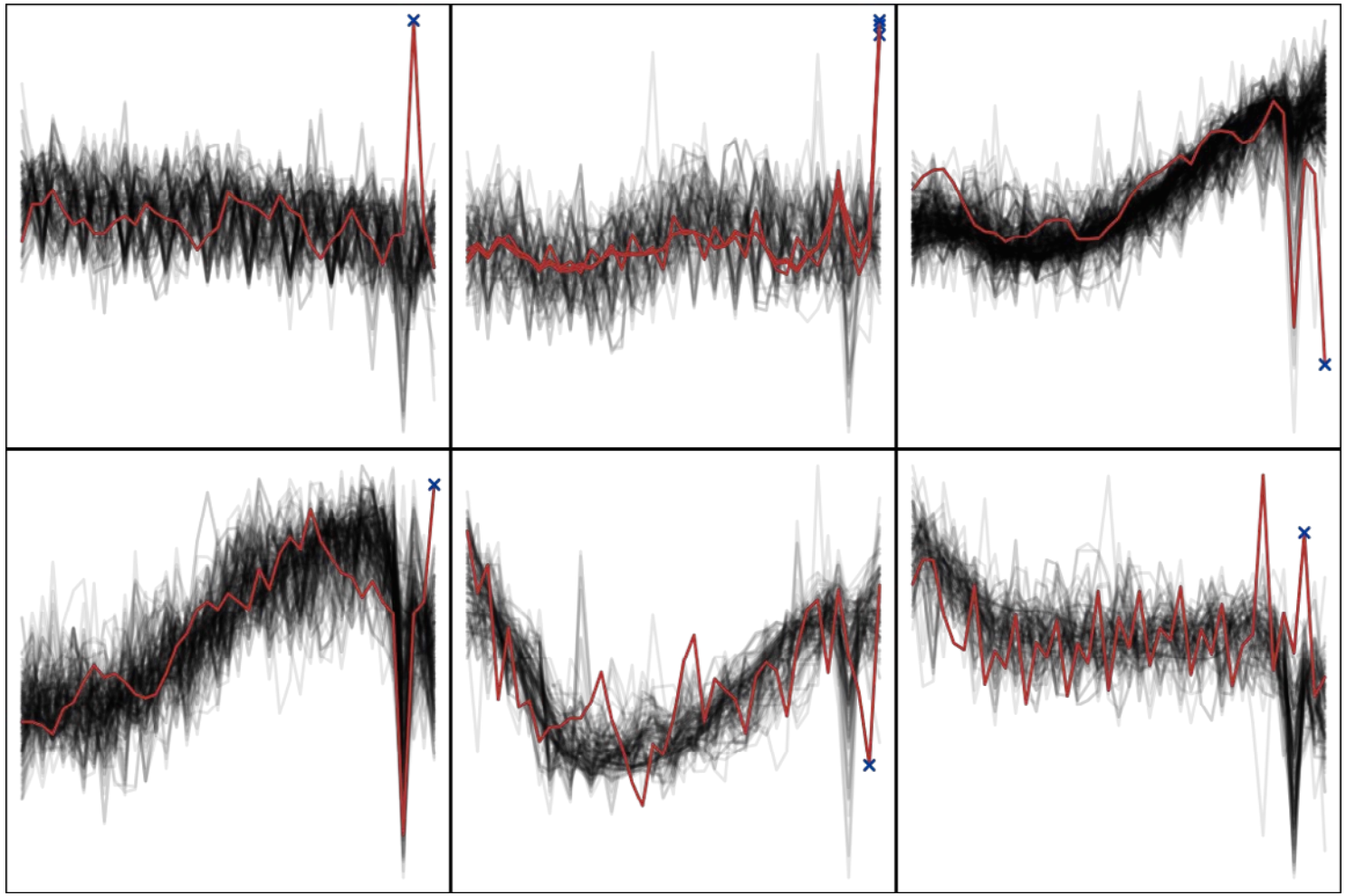


Example of 6 clusters found in the ENA data (scaled series)

Procedure (Cont'd)

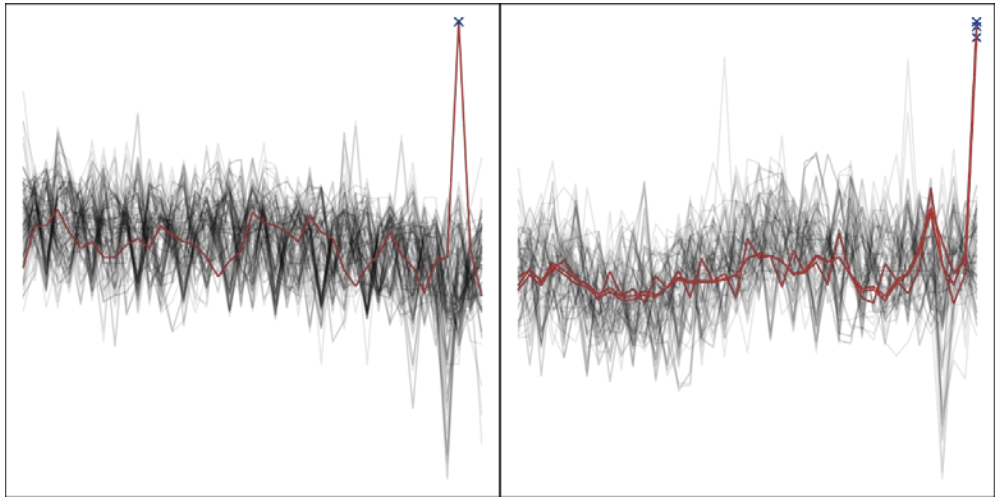
3. Detect the outlying data points in each cluster using the **DBSCAN** algorithm with the **Gower distance**

4. Investigate the flagged outliers!

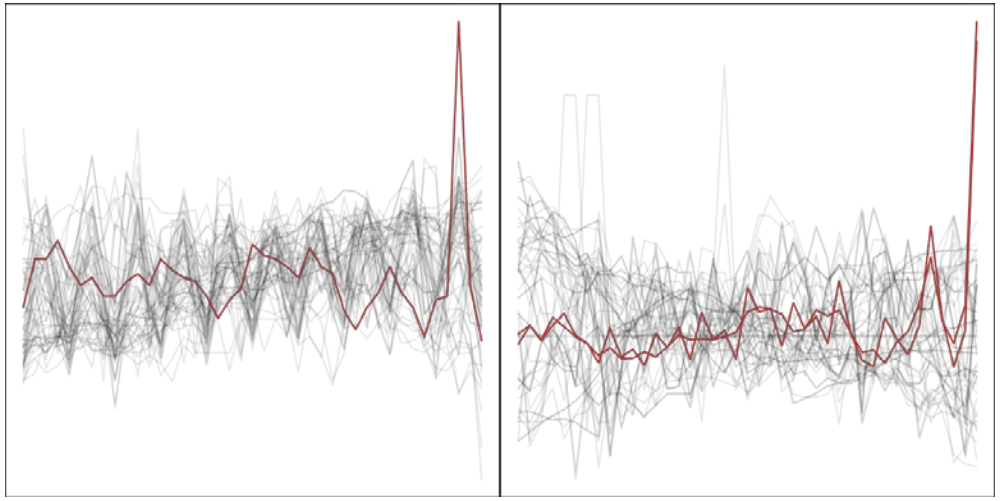


Example of 6 clusters and their outliers found in the ENA data (scaled series)

Results and Application



Example of 2 clusters and their outliers found in the ENA data (scaled series)



A and C sectors scaled series

- Used the data of the last 10 years
- 39 Clusters found
- 126 outliers found
- Spotted outliers are then investigated further and signalled to the data provider

Currently combined with a feature-additive ranking technique on the observations spotted as outliers.

Conclusion

- Outliers as observations with dynamics that differ the most from the general trend
- Robust to systemic spikes and breaks
- Data-driven, automated with few and adaptable parameters

Further improvement:

- Define the best period range to use
- Include different length time series
- Distance computation burden for Big Data cases

Thank you for your attention