Enhancing data quality controls on money market transactional data: a comparative study of anomaly detection techniques

Gianluca Boscariol[1], João Oliveira Ferreira[1], Matteo Accornero[1]

[1]*European Central Bank, Frankfurt am Main (Germany)*

Disclaimer: This paper should not be reported as representing the views of the European Central Bank (ECB). The views expressed are those of the authors and do not necessarily reflect those of the ECB.

Abstract

This paper presents four machine learning anomaly detection techniques regularly applied at the ECB to a transaction-level dataset to improve data quality, providing results on the practical effectiveness of these techniques to the Money Market Statistical Reporting (MMSR) dataset.

The anomaly detection techniques are one component of the thorough daily data quality management (DQM) applied to the MMSR dataset, among others actively searching for anomalies based on both traditional quality checks and innovative techniques. The DQM involves verifying selected transactions with reporting entities, keeping track of all feedback in a structured form, and requiring resubmission of corrected or missing data as needed.

The structure of the DQM process is essential to analyse the effectiveness of the anomaly detection techniques used, contrasting advanced machine learning tools such as isolation forests, HDBSCAN, and XGBoost against a more traditional two-step general least squares regression.

The paper concludes that XGBoost and GLS regression offer significantly higher effectiveness than the first two when applied to the MMSR dataset over the last 4 years of data, offering guidance to discriminate among innovative techniques applied to official statistics data quality.

Keywords: anomaly detection, money markets, data quality

1 Introduction

This work presents an anomaly detection project established in 2019 and used since then by the European Central Bank (ECB). The project aimed to increase the number of tools to monitor the data quality in the Money Market Statistical Reporting (MMSR). MMSR is a daily data collection that spans across four different segments of the euro money market: secured, unsecured, overnight index swap and foreign exchange swap. The data includes heterogeneous financial instruments, which are reported in a transactional representation, i.e., as flows and not as stocks. The uniqueness of MMSR data lies on its timeliness, as the data is readily available within the ECB one day after the transactions occur in the money market.

Despite the very high timeliness, MMSR data can also be considered of high quality and accuracy. This is due to a comprehensive data quality management process that analyses the transactions received both at granular and aggregated levels. The daily flow of incoming data requires a quick and coordinated action on the side of the different data quality teams involved, including the ECB and several National Central Banks (NCBs). An effective information flow and the recording of the data quality investigations conducted needs to be both quantitatively feasible, in order not to overload the reporting agents with requests, and qualitatively clear and understandable. Also considering these reasons, the use of machine learning (ML) techniques appeared as a promising solution and an ML-based project on MMSR was launched, alongside the use of more classic methods to detect errors and anomalies in the data reported.

Considering the launch in October 2019 of the euro short-term rate (€STR), the overnight unsecured rate published by the ECB, the project involving more advanced anomaly detection solutions was limited to the unsecured money market segment, to focus on transactions in a scope similar to the €STR. The daily time schedule of the workflow was also limited, to fit into the existing coordinated and broader daily data quality procedure.

Adding to the number of constraints, the IT infrastructure had a limited amount of computational capacity and memory. The final anomaly detection pipeline has taken these constraints into account and attempted to address them as best as possible.

In this paper, the anomaly detection solutions are presented, both individually and as part of a unique workflow, and compared against each other, highlighting the challenges encountered and some of the technical and operational solutions adopted.

The work is structured as follows. In the next section the MMSR dataset is briefly described and some technical remarks on the technologies involved are provided. Some main operational and organisational challenges are also highlighted, followed by an overview of the algorithms employed, completed by a few references to specific related challenges. In the Methods section, the anomaly detection pipeline is then presented and discussed, together with some technical solutions helping the efficient performance of the daily operations. Results and comparison of the different techniques are presented in the last section.

2 Theoretical Framework

2.1 Data Quality Management process

MMSR is a daily data collection on money markets' transactions involving the largest euro area banks. MMSR aims at providing the Eurosystem with a timely understanding of the functioning of euro money markets, enabling a swift and accurate monitoring of monetary policy transmission and of market expectations about policy rates future trajectories. For this reason, the MMSR data collection offers a high level of frequency, timeliness, and granularity. The data collection is conducted by the ECB with the support of several Eurosystem NCBs. Trades having been closed on one day are required to be reported to the ECB by 07:00 of the following day. Each trade eligible under the MMSR Regulation is reported on its own, not allowing for any aggregation or netting.

MMSR covers four money market segments: unsecured transactions, secured transactions, foreign exchange (FX) swaps and overnight index swaps (OIS). Transactions falling into MMSR scope have a maturity of up to and including one year. In addition, the MMSR scope is limited to trades conducted with selected non-retail counterparties, such as financial institutions (except central banks where the transaction is related to Eurosystem monetary policy operations and standing facilities), general government, and transactions with non-financial corporations classified as ‘wholesale’ pursuant to the Basel III liquidity coverage ratio (LCR) framework.

The portion of MMSR data included in this project comprises the subset of unsecured transactions closely tied to the €STR. A restriction in the amount of daily data incoming served to simplify the project’s task but did not remove the other challenges related to anomaly detection. MMSR data exhibits few highly skewed numerical features and a large number of categorical ones. The initial phase of the workflow, focused on data preparation, represented a first hurdle, in particular the challenges related to the treatment of categorical variables.

The MMSR data quality management takes place every TARGET2 opening day, involving several teams at the ECB and NCBs. This requires an efficient and structured communication and a harmonised approach to the investigation of anomalies. Organisational concerns regarding the generated workload and the feasibility of the investigations requested have therefore been part of the picture as well.

The technical infrastructure supporting the project is limited to the following components:

- Standalone R consoles for the data management and algorithms execution.
- Document management infrastructure for the structured communication and the collection of structured feedback.
- A database for the storing of logged data and investigations results.

The technical comments provided on some occasions in the following are therefore related to the R application and packages, which represent the main starting point for the more technical remarks included in the following paragraphs.

Since one of the algorithms employed in the anomaly detection process is a supervised method, a training dataset is also necessary as input to train the model. On the contrary, the other algorithms do not need this labelled dataset to build their model. In our setting the training dataset consists of a labelled anomalies dataset, containing a considerable amount of past erroneous transactions, identified as such through several data quality processes, together with correct transactions sourced from the most updated version of the MMSR database.

The construction and maintenance of a labelled anomalies database constitutes a challenge in most machine learning applications pipelines. In our setting the labelled anomalies dataset consists of two main components. A first source is represented by the centralised database containing the list of labelled anomalies identified according to the common harmonised procedures of the MMSR data quality verification work. A second source of labelled anomalies is represented by data revisions received through the MMSR data collection itself. Reported revisions related to material discrepancies are selected and added to the labelled anomalies dataset. Clusters of closely linked revisions are under-sampled to avoid more technical issues to be overrepresented.

2.2 Pre-processing (multiple correspondence analysis)

Data undergo several steps of preparation both before being used for training and before being served for daily analysis. A key aspect of the data preparation workflow is the treatment of categorical variables, given that most of the variables collected in the MMSR are of this type. Consequently, most of the information available to identify anomalies in the data has to do with non-numerical values, often representing entries of predefined lists in a standard data dictionary, such as country codes or European System of Accounts (ESA) institutional sectors.[1]

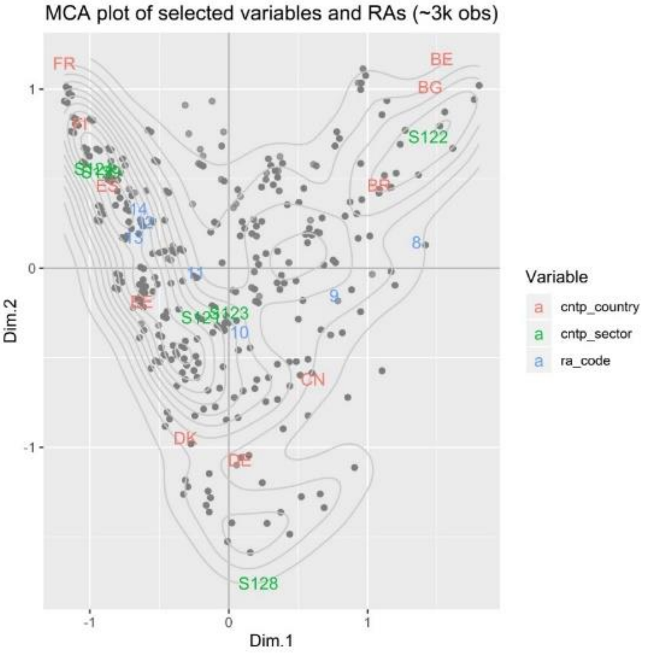
In the data preparation process adopted, the transformation of categorical variables represents a common layer, used both in the training part and for the daily anomaly detection. Before running any analysis, the categorical variables are converted into numerical variables by means of a transformation called Multiple Correspondence Analysis (MCA).[2] This transformation is applied before using the anomaly detection techniques, apart from the regression analysis.[3]

[1] See the MMSR Reporting instructions. Also see the ISO 20022 Data Dictionary, detailing data types and code lists for standardised reporting of financial and business transactions, including the MMSR messages.

[2] See Abdi and Valentin 2007 and Abdi and Williams 2010 for reference.

[3] For the regression analysis, an ad hoc clustering of levels substitutes the MCA related processing. As explained below, the regression analysis incorporates assumptions related to the functioning of money markets, which are reflected in the data preparation layer.

Figure 1: Illustration of the MCA results



Note: Illustrative analysis performed using synthetic data

MCA works as an extension of the more popular Principal Component Analysis (PCA). The starting point of the MCA is an indicator matrix, where all the levels of the involved categorical variables are converted into dummy variables. A singular value decomposition[4] is then applied to a transformation of the indicator matrix to obtain the rows and (respectively) the column factor scores. The factor scores obtained are coordinates that locate the data points (rows) and the levels (columns) in a multidimensional space. As illustrated in Figure 1 in two dimensions, data points and levels scores tend to cluster, reflecting the association or recurrence of the different features in the data.

The MCA has several advantages over alternative techniques, especially for categorical variables for which no ordinal relationship exists. This is particularly true in comparison to the one-hot and ordinal encodings.[5] The MCA row factors that substitute the original features in the analysis convey information on the relationship between the original features that are not available in the one-hot encoded dataset. Further, the MCA offers the possibility of ranking factors by variance explained. This feature, not used in the workflow described in this paper, could potentially address dimensionality issues determined by the high number of levels in the analysed features.

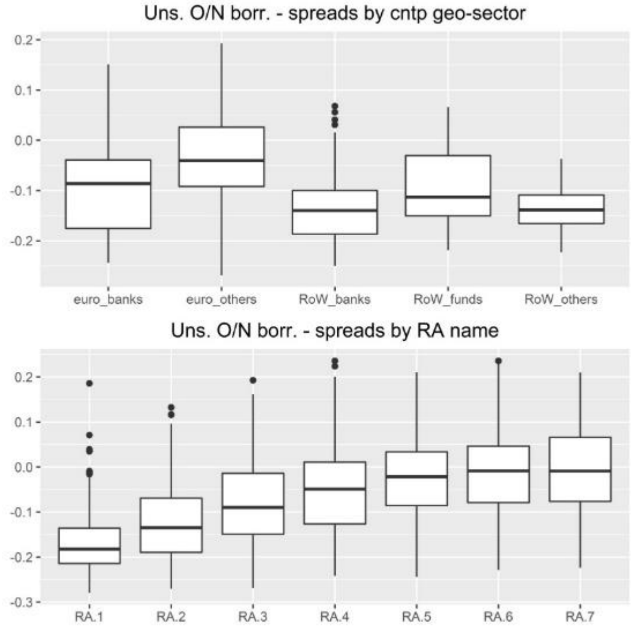
[4] See Abdi 2007 for reference.

[5] See Shyu et al. 2005 and Potdar et al. 2017 for comparisons between alternative methods.

2.3 Regression analysis

The first algorithm employed for the identification of anomalies is a linear regression analysis, used as an unsupervised learning algorithm. In fact, no information from the labelled outliers dataset is fed into the analysis. The linear regression is used to identify anomalous values with regards to a single variable: the individual transaction's deal rate. This deal rate is transformed into a spread with respect to an adequate reference rate, to allow for analysis encompassing different policy rate or market condition regimes.

Figure 2: Basic facts supporting the regression model.


Uns. O/N borr. - spreads by cntp geo-sector

Uns. O/N borr. - spreads by RA name

Note: Illustrative analysis performed using synthetic data

The spread is modelled as a function of several characteristics of the money market transaction. The characteristics relevant for the determination of the transaction's spread are identified a priori, drawing from descriptive analysis, like the one in Figure 2 and from the relevant literature on the topic.[6]

The following selected model is employed for the analysis:

$$s_i = \alpha + \boldsymbol{g}_i'\boldsymbol{\beta} + \boldsymbol{m}_i'\boldsymbol{\gamma} + \delta\log(vol_i) + \varepsilon_i$$

where s_i is the spread on ECB policy rates of transaction i, $\boldsymbol{g}_i'$ is a vector of dummy variables corresponding to all combinations of reporting agents' identifiers and macro geographical locations and economic sectors of the counterparties, $\boldsymbol{m}_i'$ is a vector of dummy variables corresponding to maturity groups, and vol_i is the transactional nominal amount of transaction i.

[6] In particular, the descriptive framework adopted is the one outlined in ECB institutional publications such as the Euro money market study (2018, 2020, 2022).

The estimation is performed employing weighted least squares.[7] Accordingly, the estimation takes place in two steps (two-steps GLS). In the first step, the residuals' estimates ($\hat{u}_i$) are obtained via an OLS regression. A multiplicative heteroscedasticity model of the form $\sigma_i^2 = \sigma^2 e^{\boldsymbol{z}_i'\boldsymbol{\alpha}}$ is then applied to obtain an estimate of the individual variances (where $\boldsymbol{z}_i$ is the vector of the covariates in the OLS for the observation i). The model is estimated by fitting: $\log(\hat{u}_i^2) = \alpha + \boldsymbol{g}_i'\boldsymbol{\beta} + \boldsymbol{m}_i'\boldsymbol{\gamma} + \delta\log(vol_i) + \varepsilon_i$. The corresponding estimates for the $\widehat{\sigma_i^2} = e^{\hat{\alpha}+\boldsymbol{g}_i'\hat{\boldsymbol{\beta}}+\boldsymbol{m}_i'\hat{\boldsymbol{\gamma}}+\hat{\delta}log(vol_i)}$ are obtained. Having defined the weight $w_i = 1/\widehat{\sigma_i^2}$ the second step is carried out with GLS, where the regression coefficients are estimated as:

$$\hat{\boldsymbol{\beta}} = \left[\sum_{i=1}^{n} w_i \boldsymbol{x}_i \boldsymbol{x}_i'\right]^{-1} \left[\sum_{i=1}^{n} w_i \boldsymbol{x}_i y_i\right]$$

The residuals e_i of the second stage regression, appropriately treated, are then used for the identification of outliers: the observations having the biggest residuals are identified as anomalies. Following Greene [2012], an appropriate transformation of the obtained residuals is performed, to avoid anomalies to 'mask' themselves, by influencing the regression results. The employed studentized residual are robust to this 'masking' effect. Having defined the influence measure $h_{ii} = x_i'\left(X_{(i)}'X_{(i)}\right)^{-1}x_i$ for each covariates vector x_i, where $X_{(i)}$ correspond to the matrix X where the observation i is excluded, and also using the standard deviation of the residuals calculated as $e'e$ where e is the vector of residuals e_i. The studentized residual $\tilde{e}_i$ is obtained as:

$$\tilde{e}_i = \frac{\dfrac{e_i}{(1-h_{ii})}}{\sqrt{\dfrac{e'e - e_i^2/(1-h_{ii})}{n-1-K}}}$$

$\tilde{e}_i$ corresponds to the weighted studentized residual for observation i, after having fitted a linear model excluding the observation i itself. Values of $\widetilde{e_i}$ particularly high in absolute value can be then used to identify anomalies in the data.

2.4 Isolation Forest

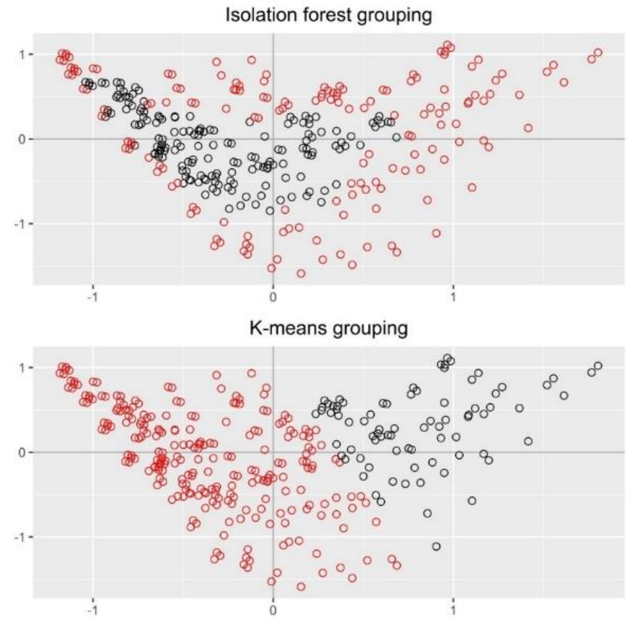
Isolation Forest is an unsupervised learning algorithm.[8] The processing of the dataset by the Isolation Forest algorithm uses a sequence of samples. Each sample contains a limited number of observations. The number of samples is also limited and defined by the user. The Isolation Forest algorithm works through each sample by means of a subsequent random sample split. At each split a feature in the dataset together with a value for its split is randomly selected. A tree is concluded with the isolation of all the observations in the sample. The

[7] See Greene 2012, chapters 4 and 9, for reference.

[8] See Liu, Ting and Zhou 2008.

process is repeated for the construction of a sequence of independent tree structures. The obtained ensemble of tree structures represents the Isolation Forest model, which can then be applied to obtain the anomaly scores. The anomaly score for a particular observation is the normalised average of the path lengths, i.e., the number of edges an instance x transverses (from root to terminal) in each tree structure in the model.

Figure 3: Isolation Forest compared to K-means



Note: Illustrative analysis performed using synthetic data

Highly efficient, the Isolation Forest algorithm is characterised by a low linear time complexity and limited memory requirements.[9] This makes the algorithm particularly suitable for efficiently processing large datasets. The preliminary sampling of the dataset helps overcoming some traditional challenges in the anomaly detection techniques, such as the identification of both scattered and clustered anomalies, and the robustness to "swamping" and "masking" effects. The Isolation Forest algorithm can be contrasted with the K-means algorithm, as shown in Figure 3. In the available cloud of data points Isolation Forest tends to isolate the periphery, irrespective of the local density and of the local clusters.

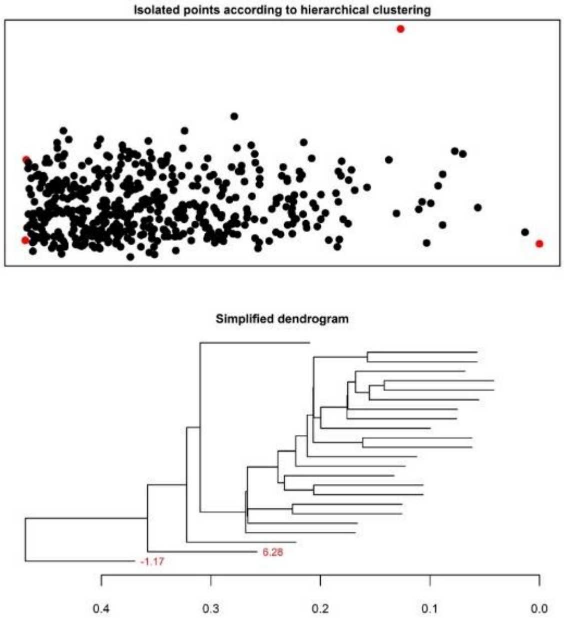
2.5 HDBSCAN

HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) is a hierarchical clustering algorithm. Hierarchical clustering is useful to identify data points isolated and poorly connected to other data points. In the present setting, HDBSCAN is thus used as

[9] See Liu, Ting and Zhou 2012 for reference.

an unsupervised learning method applied to anomaly detection. Anomalies are identified by HDBSCAN as data points isolated and poorly connected to other data points. HDBSCAN requires the user to determine the minimum size of what can be considered a set of data points (minimum cluster size).[10] HDBSCAN then works its way through the dataset, gradually lowering the threshold for the distance among data points admitted linking them into sets. Isolated data points or sets not having a minimum pre-defined number of elements above a certain threshold are gradually singled out and associated with the distance threshold at which they were excluded. As illustrated in Figure 4, additional branches of the tree are required for those observations that are more isolated from the rest.

Figure 4: Illustration of HDBSCAN outputs



Note: Illustrative analysis performed using synthetic data

The two main advantages of using HDBSCAN are the efficiency of the algorithm, which is designed to minimise time complexity[11], and the desirable integration with the anomaly score GLOSH (Global-Local Outlier Score from Hierarchies), defined as

$$\mathrm{GLOSH}(x_i) = 1 - \frac{\varepsilon_{\max}(x_i)}{\varepsilon(x_i)}$$

where $\varepsilon(x_i)$ stands for the distance threshold at which an observation x_i is singled out and excluded from the cluster and $\varepsilon_{\max}(x_i)$ stands for the lowest distance ε for which that cluster

[10] See Campello et al. 2013, Campello et al. 2015.

[11] See Hahsler et al. 2019.

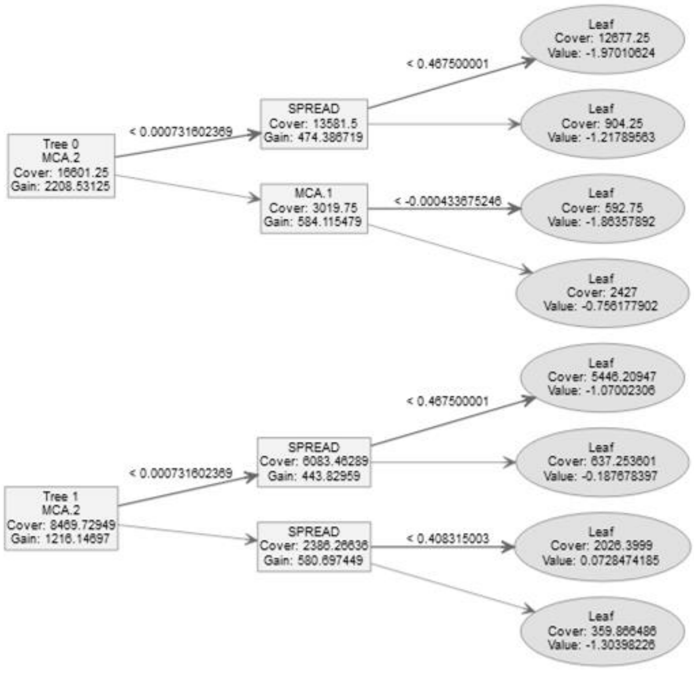
exists. In fact, GLOSH allows to factor in the anomaly score produced by the HDBSCAN algorithm the higher or lower density of the local area occupied by the evaluated observation.

A drawback of HDBSCAN (similarly to most clustering analysis) for the integration in an anomaly detection pipeline relies with the output's characteristics. HDBSCAN does not return a model applicable to distinct serving datasets. This prevents distinguishing between a training and a prediction phase in the pipeline. It also implies that the pipeline module dedicated to results' explanations has to deal with the absence of an output model to generate explanations.

2.6 XGBoost

XGBoost (eXtreme Gradient Boosting) is a very popular and successful algorithm for the solution of both classification and regression problems. Contrary to the algorithms discussed above, XGBoost is a supervised learning algorithm, requiring the labelled anomalies dataset described in section 2.1. Starting from a training sample, the XGBoost algorithm constructs a model, capturing the salient features of the data provided. During the training of the XGBoost algorithm, an ensemble of classification or regression trees is constructed, which gradually refines the prediction power of the XGBoost model.

Figure 5: Illustrative XGBoost classification trees



Note: Illustrative analysis performed using synthetic data

Figure 5 provides an illustration of how a series of classification trees in an ensemble might end up looking like after the training phase of the XGBoost algorithm, used in a classification setting. The two classification trees reported are characterised by a very limited predetermined depth (2 levels in this case). At every iteration of the XGBoost algorithm one such tree is produced to improve the model under construction. The available features and the related splits are selected based on their capability to improve the performance of the model's predictions.

Once trained, the XGBoost algorithm delivers a model that can be very efficiently applied to new data. In addition, similarly to the Isolation Forest algorithm, the model produced can be seamlessly integrated into most explanatory algorithms.

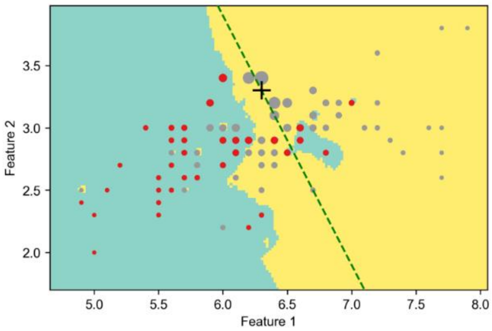
Finally, a further challenge faced by our application consisted in the hyperparameters tuning. XGBoost training requires several hyperparameters to be set in advance, like the fixed tree depth described above. The hyperparameters tuning phase is performed outside of the pipeline and aims at delivering the best results in terms of accuracy of prediction both in-sample and out-of-sample. The final purpose is the selection of the hyperparameters delivering the highest generalisation capacity of the trained model. The selected hyperparameters were obtained via cross-validation using Bayesian optimisation.

2.7 Explanation of anomalies

The algorithms employed for anomaly detection provide a list of observations that are falling outside a distribution space and are therefore classified as anomalous. To get additional insights on these anomalies we need to understand what the main drivers of the anomalous nature of these observations are.

The output from the regression analysis is quite easy to interpret since the spread is the dependent variable of the model: an unusual deal rate, compared to the deal rate distribution, is spotted for given features of a transaction and the observation is then marked as anomalous.

Figure 6: Example of LIME application



Note: Illustrative analysis performed using synthetic data

Hierarchical clustering, Isolation Forest and XGBoost are more challenging in terms of interpretability due to the higher complexity of the models and the additional data pre-processing on the input variables (grouping of categories and MCA transformation).

Among the available techniques for model explanation, the LIME approach has been chosen. LIME (Local Interpretable Model-agnostic Explanations)[12] consists of a surrogate model – usually K-LASSO or decision trees – applied on a neighbourhood of the outlying observation to explain the predictions obtained from the models above. Figure 6 illustrates how a local interpretable model represented by the dotted line is constructed for use around the instance signalled with a cross.

As LIME is a model-agnostic method, it cannot be used to make assumptions on the original model f. Therefore, a loss function $\mathcal{L}(f, g, \pi_x)$ is estimated by sampling instances z around each actual observation x. $f(z)$ will be the label for the explanation model g and $\pi_x(z)$ a proximity measure between z and x used to find $\xi(x)$ in the minimisation problem. The optimal LIME explanation can be obtained solving the following:

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \, \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

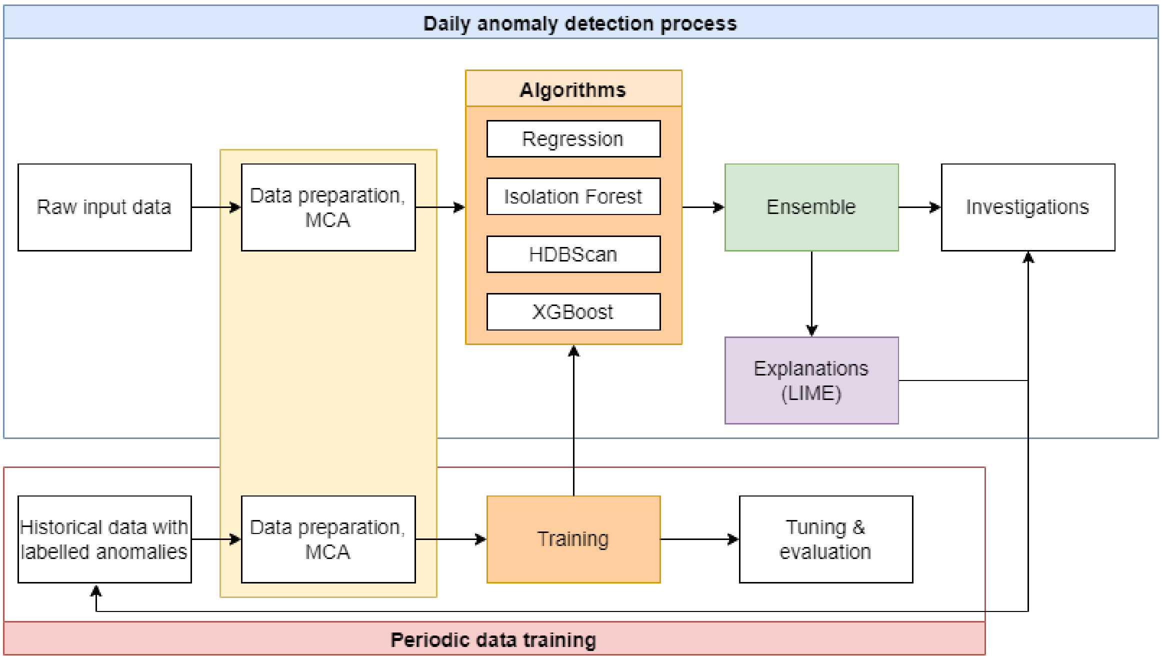
where $\Omega(g)$ is a regularisation factor as a measure of complexity, i.e., a penalty, of the explanation $g \in G$ (e.g., tree depth). The outcome of the optimisation is the most truthful interpretable model g.

[12] See Ribeiro, Singh and Guestrin 2016 for reference.

3 Methods

3.1 Anomaly detection pipeline

Figure 7: The implemented pipeline



The anomaly detection project required the design of a workflow pipeline for the daily elaboration of microdata in a limited time, with the objective of incorporating investigation results consistently and effectively into the training dataset for XGBoost. The key steps of the pipeline can be defined as follows:

- **Data preparation** (in yellow in Figure 7). This layer was needed to adapt the input data to be used in the elaboration phase, both in terms of data structure and in terms of data distributions.[13] The most impacting transformation involves data enrichment, MCA pre-processing of categorical variables and creation of calculated fields to reduce seasonality effects of the rates.
- **Application of algorithms** to the transformed input data, or training of the supervised one (in orange in Figure 7). While the regression and the two unsupervised methods do not need a training phase, in the case of XGBoost a periodic training is necessary to update the model with the most recent data.

[13] See Polyzotis 2018, for the identification of the most common concerns regarding the correspondence between training and serving data. The conceptual distinction in the ML pipeline and workflow are mainly drawn from Polyzotis 2018, Arpteg 2018 and Crankshaw 2015.

- **Ensemble** (in green in Figure 7). This step collects the output of the previous phase, with the aim of filtering the most promising anomalies from the top scoring observations identified by the four anomaly detection algorithms.
- **Explanation** (in blue in Figure 7). This step produces interpretable explanations for the results of three out of the four algorithms – in the case of regression, all anomalies refer to anomalous deal rates.
- **An overall feedback loop** to timely and automatically feed investigations' results into the training step for XGBoost and to serve as a benchmark for the evaluation of unsupervised learning methods.

The implementation phase of the project, leading to the set-up of the pipeline depicted in Figure 7, revealed several challenges. Most of them were related to the use of MCA as encoding technique of categorical variables in the data preparation layer. The challenges encountered reflected in high elaboration time issues, which were particularly critical in a framework foreseeing a timely daily production of outputs to be transmitted to the verification teams early each morning. In addition, explanation of results was hindered by the MCA transformation. More in detail, these were the most relevant challenges encountered:

- To ensure a homogeneous data transformation for both training and serving steps, while keeping the elaboration times low, the information made available by the MCA in the training phase had to be modified and optimised.
- To ensure a readability of the results of the explanatory algorithms further information related to the MCA transformation had to be pushed down the pipeline up to the Explanation step.

The Ensemble step played a key role to keep results relevant and the related verification work feasible.

In the next section, the solutions identified for the highlighted problems are presented and commented in detail, referencing back to the underlying algorithms, and proving the rationale of the code added to the traditional libraries employed.

3.2 Data preparation and MCA objects manipulation

The described pipeline features a data preparation layer for both training and serving dataset which includes MCA as data transformation algorithm for the involved categorical variables. At the time of execution of the serving data preparation and analysis, the following information about the MCA transformation is required:

- The MCA model: a series of matrices containing the necessary information to perform the transformation of the raw input data.
- Detailed information on the levels of the variables to be transformed, to make the transformations previously applied to the training dataset reproducible.
- Information tracing back the obtained MCA factors to the original dataset variables.

Most popular off-the-shelf packages[14] for the performance of MCA transformations provide the required inputs but, having a general purpose, are not optimised to deal with the pipeline described and require technical solutions adopted to optimise the data preparation using MCA are exposed and commented.

A first intervention dealt with point 1 above and aimed at reducing the dimension of MCA objects. In the pipeline described, in fact, MCA objects need to be saved after the elaboration in the training step to be recalled during the serving data preparation step. Off-the-shelf MCA functions applied to datasets having dimensions $n \times k$ (n being the number of instances and k the number of categorical variables) produce MCA objects including matrices preserving the full dataset length n, thus increasing the time necessary for loading these objects. This feature of the algorithms becomes critical when facing large datasets with constrained computation resources.

The solution identified for this problem relied on the fact that in a typical anomaly detection pipeline employing MCA, only the row factors scores are used for analysis. The important information to be stored and to be made available to the serving dataset preparation refer to the inputs necessary to obtain the row factors and can ignore any input not strictly related to this purpose, among others the derivation of the columns' factors. Having this consideration in mind, the solution obtained used the conversion formula defined by Abdi and Williams[15] to solely store the matrices useful to obtain the row factors scores in the serving dataset and thus avoiding any n-dimensional matrix to be saved together with the model.

The solution obtained relied on the following MCA algebraical properties. The starting point for an MCA transformation is an $n \times k$ dataset featuring categorical variables. Having defined:

- $\boldsymbol{X}$ as the $n \times j$ indicator matrix, where all k categorical variables are already converted to j dummies,
- N the sum of all $\boldsymbol{X}$ elements,
- $\boldsymbol{Z}$ as $N^{-1}\boldsymbol{X}$,
- $\boldsymbol{r}$ and $\boldsymbol{c}$ as the row totals and the column totals of $\boldsymbol{Z}$,

[14] The R packages tested and employed have been MASS and FactoMineR.

[15] See Abdi and Williams 2010.

- $\boldsymbol{D_r}$ as a $n \times n$ matrix and $\boldsymbol{D_c}$ as a $j \times j$ matrix, as $\boldsymbol{D_r} = \mathrm{diag}(\boldsymbol{r})$ and $\boldsymbol{D_c} = \mathrm{diag}(\boldsymbol{c})$, having performed the following Singular Value Decomposition (SVD):

$$\boldsymbol{D}_{\boldsymbol{r}}^{-\frac{1}{2}}(\boldsymbol{Z} - \boldsymbol{rc}')\boldsymbol{D}_{\boldsymbol{c}}^{-\frac{1}{2}} = \boldsymbol{P\Delta Q}'$$

where $\boldsymbol{P}$ is a $n \times\ j$ matrix, $\boldsymbol{\Delta}$ a $j \times j$ diagonal matrix containing the singular values of the left-hand side expression, and $\boldsymbol{Q}'$ a $j\ \times j$ matrix,

row factor scores $\boldsymbol{F}$ are obtained as:

$$\boldsymbol{F} = \boldsymbol{D}_{\boldsymbol{r}}^{-\frac{1}{2}}\boldsymbol{P\Delta}$$

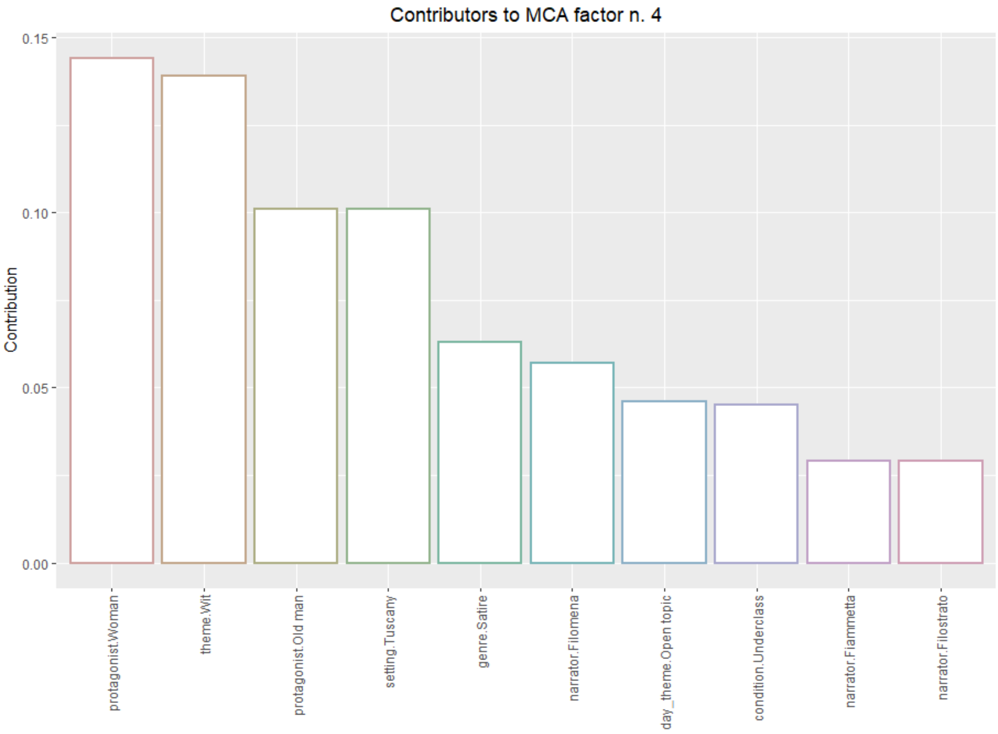
and column factor scores $\boldsymbol{G}$ are obtained as:

$$\boldsymbol{G} = \boldsymbol{D}_{\boldsymbol{c}}^{-\frac{1}{2}}\boldsymbol{Q\Delta}$$

Instead of relying on the n-dimensional matrix $\boldsymbol{P}$, the solution obtained exploits the conversion formula and obtains the rows factor scores for the serving dataset rows relying on $\boldsymbol{G}$, which has $j \times\ j$ as dimensions. This means that $\boldsymbol{G}$ is a far preferrable object to save and recall when dealing with large datasets. The row factor scores for new observations can be obtained as:

$$\boldsymbol{f}_{\mathrm{new}} = (\boldsymbol{i}'_{\mathrm{new}}\mathbf{1})\boldsymbol{i}'_{\mathrm{new}}\boldsymbol{G\Delta}^{-\mathbf{1}}$$

Figure 8: Feature contribution to an individual MCA component

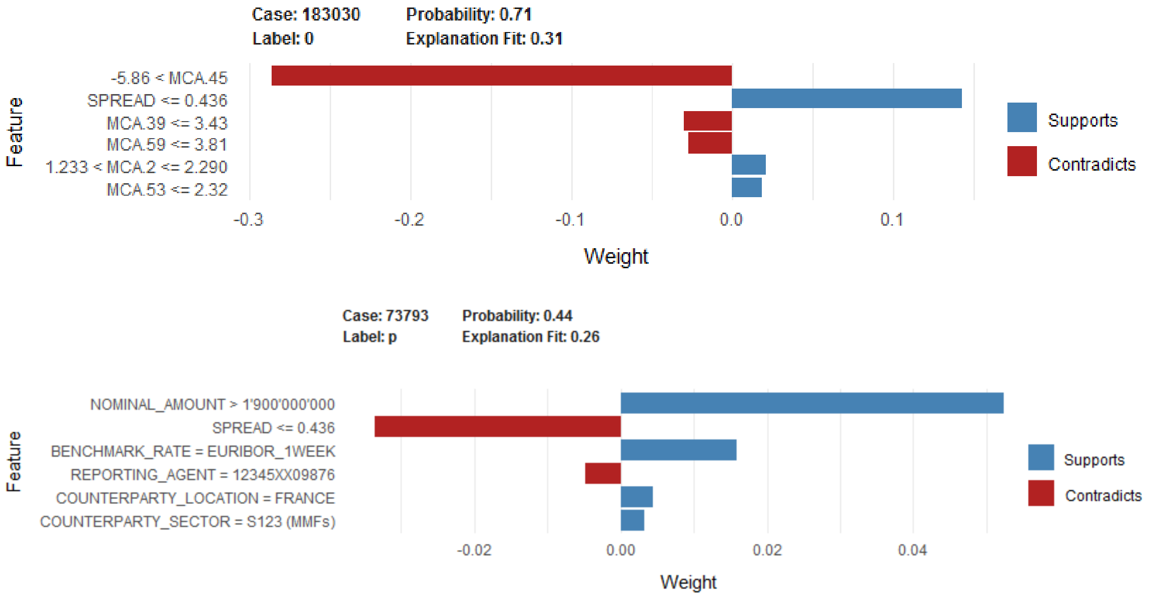


Note: Illustrative analysis performed using synthetic data

This solution speeded up considerably the saving and loading processes related to MCA models. An application using an example dataset of approximately 50,000 records and 40 levels showed a decrease in size of the saved objects of -95% (3.5 vs 77 kB).

A second intervention dealt with the inclusion in the MCA data model of inputs usable to tackle points 2 and 3 above. In any data preparation step, the obtained MCA model can be applied to a new dataset only if the latter can be first transformed into a compatible matrix $\boldsymbol{X}$. The most efficient solution identified for this problem consisted in storing, together with the MCA model, information related to the structure of the original dataset, to be employed to perform a compatible transformation of the new incoming dataset before the application of the MCA model. Further, this solution dealt with the possible presence of new levels in the incoming dataset by explicitly introducing into the MCA model a residual column for non-modelled levels.

Figure 9: Illustrations of LIME output with and without MCA factors included in the model.



Note: Illustrative analysis performed using synthetic data

Finally, contribution coefficients were stored to identify the levels (and the related variables) possessing a high importance for a determined MCA factor. As exemplified in Figure 7, contribution coefficients can help understand the underlying factors and variables for each MCA factor in a graphical way. They can also be fed into further layers of the pipeline to obtain explanations referring to the original variables, and not to one MCA factor itself, as explained below.

3.3 Interpretation and ensemble

The pipeline developed included a step dedicated to the explanation of anomalies with the purpose of simplifying the investigation tasks. The challenges encountered in this area of the pipeline were mainly related to:

- MCA factors are of little use when included in the explanation of a raised anomaly. A treatment of the explanation had to be foreseen to trace these factors to the original features.
- LIME off-the-shelf packages typically produce a complex and rich output (like the one exemplified in Figure 9) that better suits a graphical representation rather than supporting a textual reference. The necessity of an efficient communication and storage of information suggested the production of a numerical and textual synthesis of the standard LIME output.
- The LIME algorithm could be directly applied only to a subset of the algorithms employed, partly due to technical limitations and partly due to methodological constraints. Standard wrappers for enveloping algorithms outputs were therefore elaborated.

To overcome the issues reported, an extension of the LIME package was developed. This extension extracts the features encapsulated in the MCA components and subsequently weighs them by their contribution to the MCA factor and by the LIME weight attributed to the same MCA factors. Only those MCA factors supporting the outlier hypothesis were kept making the message more straightforward. With the data obtained, it is possible to reconstruct the relative weight of each feature and group them in macro-categories readily understandable for the human investigation. A toy example is provided in the tables below.

Table 1: Illustrations of LIME output with and without MCA factors included in the model.

	MCA 1	**MCA 2**	**MCA 3**
Feature A	50 %	10 %	10 %
Feature B	30 %	60 %	10 %
Feature C	20 %	30 %	80 %
LIME weight	0.02	-0.1	0.3

Note: Illustrative analysis performed using synthetic data

Table 2: Relative weights of dataset features

Feature A	Feature B	Feature C
12.50%	11.25%	76.25%
Category 1		Category 2

Note: Illustrative analysis performed using synthetic data

In the case of clustering, no model can be provided for explanation of the LIME algorithm. In our specific case, HDBSCAN results could not be directly explained using LIME. The problem was solved by training and fitting a “dummy” XGBoost model on the data used by HDBSCAN setting the label to 1 or 0 in accordance with the anomaly flag. With this, a LIME-compatible model could be obtained, representing with great accuracy the results obtained with HDBSCAN.

All the above-mentioned steps were executed as part of a production activity with the time constraints recalled above. For this reason, to save on execution time, LIME was applied only on selected outlying transactions extracted from the results of the different algorithms based on their score. This score could either be a probability measure or value of residuals, depending on the model applied. The selection of anomalies taking place in the Ensemble step limited the execution time for the Explanation step and ensured the feasibility and relevance of the identified anomalies.

In the Ensemble step, the full set of observations is filtered to obtain a small group of transactions that will be part of the daily data quality investigation. A comparison against the historical database of labelled anomalies is part of the Ensemble process: if previous correct observations were found to be very similar to the chosen outliers, these were removed from the list and not investigated. Additionally, only the two top-scoring observations from each algorithm and the globally top four ordered by impact are kept. The impact measure is trivially computed by considering the product between the transactional nominal amount and the trade rate spread.

4 Results

4.1 Dataset description

Advanced anomaly detection techniques are applied to a subset of the MMSR data consisting of unsecured deposits transactions, which represent most of the unsecured segment transactions.

The historical data dataset used for training purposes is characterised by a set of descriptive features that include:

- **Reporting agent LEI:** Legal Entity Identifier of the reporting agent.
- **Counterparty sector:** Sector classification of the counterparty.
- **Counterparty location:** Location of the counterparty, grouped into macro areas to reduce cardinality of the feature.
- **Maturity bucket:** Identifier for the maturity bucket into which the transaction falls – a maturity bucket contains a range of days-to-maturity values.
- **Transactional nominal amount:** The nominal amount of the transaction.
- **Deal rate spread:** The spread value of the transaction, calculated as the difference between the deal rate and a relevant reference rate.

The training dataset contains approximately 1.5 million rows, starting from the year 2019, providing a substantial volume of data for analysis. Anomalies represent a minor share of the dataset, accounting for only 0.03% of the observations. This indicates a highly imbalanced dataset with the anomaly class being very infrequent, a characteristic that was taken into consideration when developing predictive models.

4.2 Performance analysis

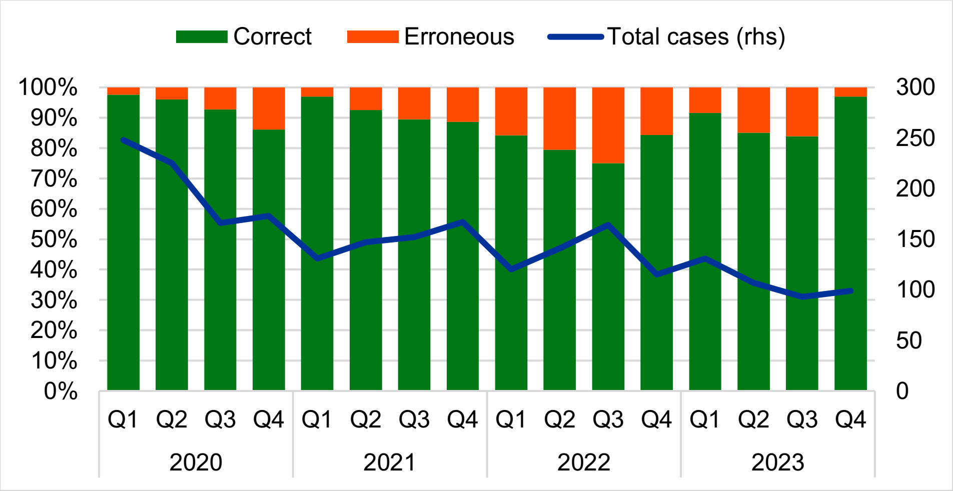
There are several aspects that add complexity to the assessment of out-of-sample performance for the four anomaly detection techniques on a day-to-day production scenario:

- The maximum number of daily investigations is capped to four transactions in total and two for each algorithm due to workload constraints. This poses the question on how to select possible anomalies among the outputs of different methods.
- During periods with very low or inexistent anomalies in the reported data, which are quite frequent, the scores assigned by each algorithm to the daily observations are generally very low. Hence, selecting outliers becomes a nearly impossible task if we were to follow the approach of setting a minimum threshold value for the scores. Our choice was instead to restrict the investigation to the two highest-scoring observations for each algorithm in each day, with a maximum number of four investigations. This ensures a steady flow of investigations but undeniably diminishes the algorithms' overall measured accuracy.

- The final scores assigned by each algorithm to the transactions have different distributions and magnitudes. Therefore, even after normalisation, they still carry relevant differences which make the scores difficult to compare across the four methods. In addition, the regression analysis returns residual values instead of proper scores.

Due to the above constraints and challenges, it was observed that in general it is quite difficult to avoid false positives with the current setup. The main reason is that, since a fixed threshold for the scores cannot be easily identified, the cap at four observations per day allows also for correct transactions to enter the set of daily requests, even on those days which do not have highly suspicious reported transactions (see Figure 10).

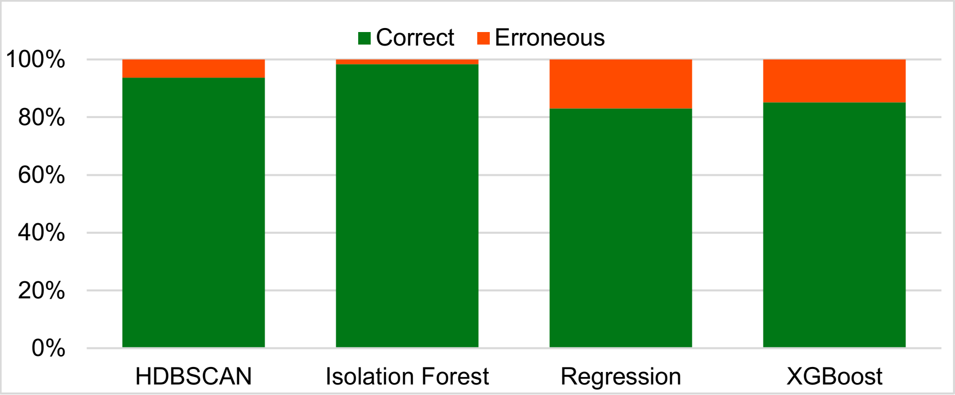
Figure 10: Quarterly share of confirmed erroneous trades over total investigated trades (lhs) and number of total investigated trades (rhs)



For these reasons it is not possible to use standard concepts such as sensitivity and specificity to measure the accuracy of each anomaly detection technique. Rather the approach is to perform a comparative analysis of the results obtained in the application of these techniques under the conditions described above.

To assess the performance of the four techniques, the share between investigated trades confirmed as erroneous by the Reporting Agent and the total of investigated trades will be used. The results are shown in figure 11.

Figure 11: Share of investigated trades grouped by investigation result and algorithm.

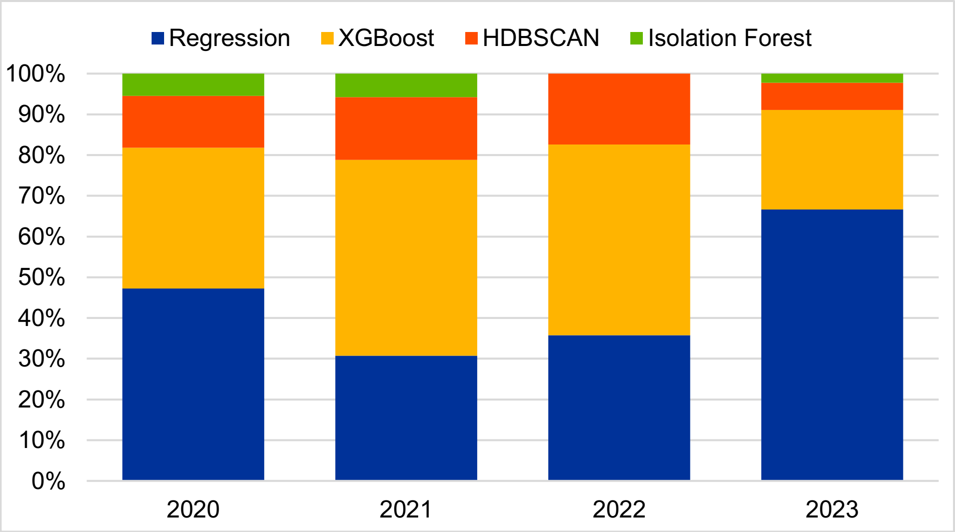


Among the four anomaly detection techniques, the regression analysis (model based) and XGBoost (supervised) have consistently outperformed the two unsupervised methods.

Isolation forest is the worst of the four, with about 1.7% only of confirmed errors out of the total anomalies selected, i.e., cases assessed as positive, while HDBSCAN performed better with 6.3%. XGBoost and regression score better with 15% and 17% respectively of positively tested cases confirmed as errors.

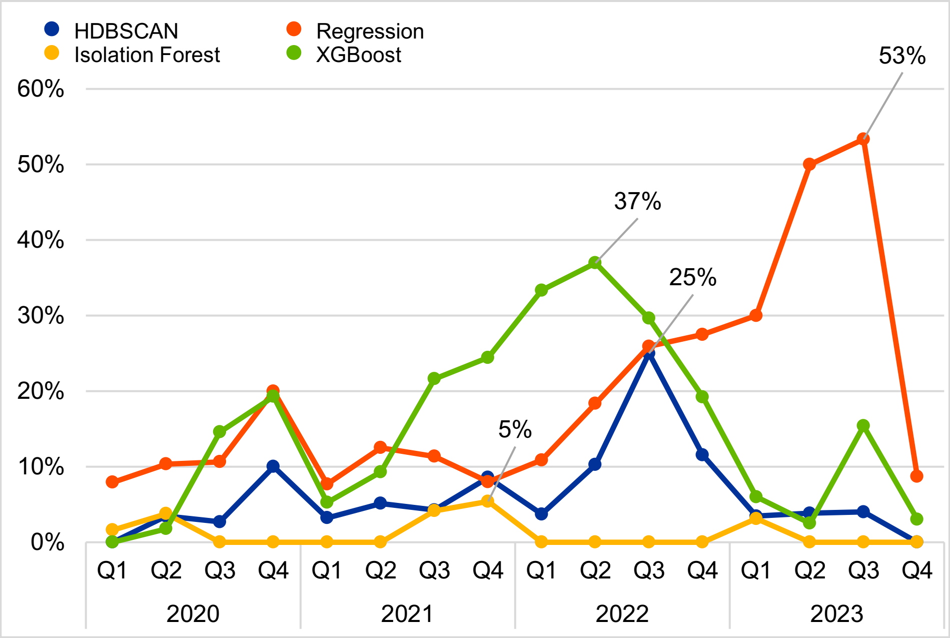
Focusing exclusively on the confirmed anomalies, XGBoost and regression together always account for at least 80% of the investigated cases, as shown on figure 12. The remaining cases are prevalently result of the HDBSCAN unsupervised algorithm, while a few cases are coming from the Isolation Forest.

Figure 12: Percentage of identified erroneous trades, grouped by algorithm.



The quarterly time series for anomaly detection accuracy in Figure 13 reveals the algorithms' responsiveness to the ECB rate hikes, which started in the summer of 2022. The regression algorithm has become progressively more reliable in identifying anomalous transactions since then, achieving a quarterly accuracy that exceeds 50% during the second and third quarters of 2023. The investigations in previous erroneous trades in these quarters had a positive impact on the accuracy of the reporting, which in turn caused a lower number of erroneous trades being report for the subsequent quarter – explaining the decrease in erroneous cases selected from all algorithms.

Figure 13: Quarterly share of investigated cases with outcome "Erroneous" by algorithm



The XGBoost anomaly detection algorithm does not achieve the same high performance as the regression. The peak accuracy of 37% was obtained in the second quarter of 2022 after a series of increasingly performing quarters. However, since then it has been declining, a matter currently being investigated.

Re-trainings of the XGBoost algorithm are performed at a rather low frequency to allow enough new data to be available. The usual training frequency ranges between 6 to 9 months. Following each re-training, a slow but consistent decline in in-sample performance has been observed. Specifically, the AUC-PR of the test set now stabilises at approximately 80%, which contrasts with the higher 85% achieved in the past. HDBSCAN and Isolation Forest have performed worse than the XGBoost and Regression techniques.

5 Conclusions

This paper introduces, discusses, and compares a compact workflow tailored for anomaly detection within the MMSR dataset. We provide a comprehensive overview of the anomaly detection pipeline, focusing on technical and methodological strategies for optimising data preparation.

Among the four anomaly detection techniques, XGBoost and GLS regression emerged as the most promising and accurate. On the other hand, unsupervised methods such as Isolation Forest and HDBSCAN exhibited worse performance. Our findings suggest that supervised techniques offer relevant advantages over unsupervised techniques, and simpler methods like the GLS regression are sufficiently effective to compete with supervised methodologies in the setting of the MMSR dataset.

Looking forward, we identified several potential improvements for our pipeline, including leveraging the available labelled data more effectively, prioritising research on newer supervised techniques, and refining the current training dataset by involving the use of yield curves to better categorise the trades duration.

References

Abdi, H. (2007). Singular Value Decomposition (SVD) and Generalised Singular Value Decomposition (GSVD). In N.J. Salkind (Ed.), *Encyclopedia of Measurement and Statistics*, Sage, Thousand Oaks.

Abdi, H., & Valentin, D. (2007). Multiple Correspondence Analysis. In N.J. Salkind (Ed.), *Encyclopedia of Measurement and Statistics*, Sage, Thousand Oaks.

Abdi, H., & Williams, L. J. (2010). Correspondence Analysis. In N.J. Salkind (Ed.), *Encyclopedia of Measurement and Statistics*, Sage, Thousand Oaks.

Arpteg, A., Brinne, B., Crnkovic-Friis, L., & Bosch, J. (2018). Software Engineering Challenges of Deep Learning. In *44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2018, pp. 50-59.

Campello, R. J. G. B., Moulavi, D., & Sander, J. (2013). Density-Based Clustering Based on Hierarchical Density Estimate. In *Proceedings of 17th Pacific-Asia Conference*, PAKDD 2013 Gold Coast, Australia, April 2013.

Campello, R. J. G. B., Moulavi, D., Zimek, A., & Sander, J. (2015). Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection. In *ACM Transactions on Knowledge Discovery from Data*, 10(1).

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Knowledge Discovery from Data* '16, August 13-17, 2016, San Francisco, California, USA.

Crankshaw, D., Bailis, P., Gonzalez, J. E., Li, H., Zhang, Z., Franklin, M. J., Ghodsi, A., & Jordan, M. I. (2015). The Missing Piece in Complex Analytics: Low Latency, Scalable Model Management and Serving with Velox. In 7th Biennial Conference on Innovative Data Systems Research (CIDR '15) January 4-7, 2015, Asilomar, California, USA.

Greene, W. H. (2012). *Econometric Analysis* (7th Ed.). Pearson Education Limited, Harlow - Essex.

Hahsler, M., Piekenbrock, M., & Doran, D. (2019). DBSCAN: Fast Density-Based Clustering with R. *Journal of Statistical Software*, 91(1), 1-30.

Hastie, T., Tibshirani, R., & Friedman, J. (2017). *The Elements of Statistical Learning* (2nd Ed.). Springer, New York.

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2012). Isolation-Based Anomaly Detection. In *ACM Transactions on Knowledge Discovery from Data*, 6(1).

McInnes, L., Healy, J., & Astels, S. (2019). HDBSCAN. Documentation Release 0.8.1. https://readthedocs.org/projects/hdbscan/downloads/pdf/latest/ .

Polyzotis, N., Roy, S., Euijong Whang, S., & Zinkevich, M. (2018). Data Lifecycle Challenges in Production Machine Learning: A Survey. *SIGMOD Record*, 47(2).

Potdar, K., Pardawala, T. S., & Pai, C. D. (2017). A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers. In *International Journal of Computer Applications*, 175(4).

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*.

Shyu, M. L., Sarinnapakorn, K., Kuruppu-Appuhamilage, I., Chen, S.-C., Chang, L. W., & Goldring, T. (2005). Handling Nominal Features in Anomaly Intrusion Detection Problems. In *15th International Workshop on Research Issues in Data Engineering: Stream Data Mining and Applications* (RIDE-SDMA'05). IEEE Computer Society, Washington DC.