



BANK ĊENTRALI TA' MALTA
EUROSISTEMA
CENTRAL BANK OF MALTA

CENTRAL BANK OF MALTA WORKING PAPER



BANK ĊENTRALI TA' MALTA
EUROSISTEMA
CENTRAL BANK OF MALTA

Uncovering Anomalies in the Granular Securities Holdings Data of Maltese Banks: An Ensemble Modelling Approach using Unsupervised Machine Learning Techniques¹

Giulia Maria Tabone²

WP/09/2026

¹I would like to thank Prof David Suda for reviewing an earlier version of this paper and suggesting areas of improvement. I would also like to thank Mr Jesmond Pule and Ms Karen Caruana for their guidance and helpful feedback. I am also grateful for the opportunity to present this paper at the RSC Conference 2026 in Malta and would like to thank the participants for their valuable feedback and constructive comments. I also wish to thank the members of the Machine Learning and Big Data Network of the Central Bank of Malta for their comments and feedback.

The views expressed in this paper are the author's own and do not necessarily reflect the views of the Central Bank of Malta. Any queries should be sent through the following Contact Us form.

²Senior Economist Statistician, Statistics Department, Central Bank of Malta

Abstract

This paper presents a semi-automatic machine learning framework for detecting anomalies in Granular Securities Holdings Data of Maltese banks which have an ISIN, aggregated over the total bank holdings. The proposed approach leverages time series analysis techniques to identify irregular patterns and deviations in this data, with the aim of enhancing data quality. Three anomaly detection methods, namely, Isolation Forest (iForest), One Class-Support Vector Machines (OC-SVMs) and Hierarchical DBSCAN (HDBSCAN), are employed. An ensemble modelling approach based on majority voting integrates these methods in an attempt to improve robustness and better the quality of the anomalies identified. The framework is fully implemented in Python, with results demonstrated through tables.

Contents

Abstract	ii
Contents	iii
List of Abbreviations	v
1 Introduction	1
2 Literature Review	2
3 The Dataset and Methodology	4
3.1 The Dataset	4
3.2 Methodology	5
3.2.1 Data Preprocessing	5
3.2.2 HDBSCAN Parameter Selection	5
3.2.3 Isolation Forest and One-Class SVM	6
3.2.4 Model Combinations and Ensemble Approach	7
3.2.5 Model Evaluation and Comparison	8
4 Results and Discussion	9
4.1 Computational Performance	10
4.2 Comparison of Parameter Combinations	10
4.3 Temporal Behaviour of the Final Anomaly Rate	11
4.4 Comparison with Classical Methods	13
4.5 Discussion	16
5 Concluding Remarks	18
A Statistical Theory	21
A.1 Sliding Window Approach	21
A.2 Isolation Forest	21
A.3 One Class Support Vector Machines	23

A.4 HDBSCAN	25
References	29

List of Abbreviations

IQR Inter-Quartile Range

iForest Isolation Forest

iTree isolation tree

BST binary search tree

SVM Support Vector Machine

OC-SVM One Class-Support Vector Machine

RBF radial basis function

LOF Local Outlier Factor

AE Auto-Encoder

LSTM Long-Short Term Memory

ARIMA Autoregressive Integrated Moving Average

DBSCAN Density Based Spatial Clustering of Applications with Noise

HDBSCAN Hierarchical DBSCAN

GESD Generalized Extreme Studentized Deviate

AIC Akaike Information Criterion

ADF Augmented Dickey-Fuller

1 Introduction

Prior to analysing any data and disseminating, it is important to confirm the data's quality, as more often than not, the data may contain anomalies. An *anomaly* is an observation that exhibits unusual behaviour when compared to the rest of the data. This deviation can represent a number of scenarios, from significant events, to human errors, emerging trends and others. The process of identifying unusual patterns that do not conform to the expected behaviours can thus be defined as *anomaly detection*.

The objective of this study is to develop a framework capable of detecting anomalies in the Granular Securities Holdings Data of Maltese Banks. After identifying said anomalies, the relative bank is queried to confirm whether the anomaly is in fact an error or to provide more information on the movements which is then used for analysis. The Granular Securities Holdings Data of Maltese Banks dataset (more information in Section 3) is time series in nature, and so it is essential to utilize methods capable of handling this type of data. In the time series literature, anomalies are typically classified into three types, namely, *point anomalies*, *contextual anomalies* and *collective anomalies*.

A point anomaly is an individual data point that is too far off from the other data points and occurs when a particular observation differs significantly from the data distribution. Point anomalies resemble what are conventionally referred to as “outliers”. These type of anomalies do not take into consideration the context or sequence of a time series. Contextual anomalies are observations that deviate significantly from the rest of the data within a specific context, however, outside that context, these data points are consider to be normal. Lastly, collective anomalies are a group of data points that, when considered together, deviate significantly from the expected pattern, even though the individual data points may not be considered anomalies on their own. Contextual and collective anomalies do take into account the context or sequence of a time series.

Anomaly detection is a challenging task, particularly because it is often carried

out in an unsupervised setting, where unusual behaviour must be identified without labelled examples. This becomes even more difficult for time series data, which may exhibit noise, temporal dependence, non-stationarity, and changing patterns over time. Classical approaches, such as Z -score methods, Grubbs’ test, and its generalisations, provide simple and computationally efficient anomaly detection tools, but their applicability may be limited when assumptions concerning the statistical properties of the data, such as normality or stationarity, are not satisfied. For this reason, the proposed framework adopts an automated unsupervised ensemble combining Isolation Forest, One-Class Support Vector Machines, and HDBSCAN, which provides a more flexible, data-adaptive approach to anomaly detection. Classical methods are subsequently included as simple baseline approaches for comparison with the machine-learning framework.

The remainder of this paper is organised as follows. Section 2 provides a review of the literature on anomaly detection methods for time series data. Section 3 describes the dataset and methodology. The results are presented and discussed in Section 4, followed by concluding remarks in Section 5. Additional theoretical material is provided in the Appendix.

2 Literature Review

Within the time series literature, numerous techniques have been developed to identify anomalies. Conventionally, classical statistical approaches such as the Z -score method, the Inter-Quartile Range (IQR) method and Generalized Extreme Studentized Deviate (GESD) were widely used. The Z -score method measures how far a data point lies from the mean in standardized units and flags observations that exceed a pre-defined threshold. However, time series are often non-stationary, meaning normal temporal fluctuations may be misidentified as anomalies, leading to false positives (Shakil et al., 2023). Additionally, the Z -score assumes normality and is highly sensitive to extreme observations because it uses the standard deviation as its scale estimator (Yaro et al., 2024). To address these limitations, modified Z -score methods based on the median absolute deviation or S_t scale estimators were proposed (Wilcox, 2002; Yaro et al., 2023). The IQR method treats observations lying beyond the upper and lower bounds defined by the interquartile range as out-

liers, however, its effectiveness declines in non-normal or highly skewed distributions (Chandola et al., 2009). GESD detects anomalies by iteratively evaluating whether the most extreme observation’s test statistic exceeds the critical value under an assumed normal distribution (Rosner, 1983). The limitations associated with classical approaches have motivated the shift towards more flexible, data-driven statistical learning techniques.

A variety of such methods have been explored for time series anomaly detection. Auto-Encoders (AEs) learn latent representations of the input data and attempt to reconstruct the original signal and observations whose reconstruction errors exceed a specified threshold are labelled anomalous. Different AE variants exist, however, the classical and variational AEs are the most common for anomaly detection. For sequential datasets, recurrent AEs built using Long-Short Term Memories (LSTMs) or Bidirectional LSTMs are commonly adopted because they preserve temporal dependencies, with Seq2Seq autoencoders offering further flexibility. Although these models have demonstrated strong performance in financial anomaly detection and stock-market irregularity analysis (Yang & Liu, 2025), they require long sequences, relatively clean training data, extensive hyperparameter tuning, and can be computationally intensive (Neloy & Turgeon, 2024).

Density-based methods provide additional alternatives. The Local Outlier Factor (LOF) measures how much the local density around each point deviates from the densities of its neighbours, flagging points with significantly lower density as outliers. LOF performs well in datasets with varying local densities, making it effective when anomalies are locally isolated rather than globally distinct (Breunig et al., 2000). Nonetheless, its performance deteriorates in high-dimensional settings, and it requires careful tuning of the neighbourhood parameter. Computational complexity also increases notably with dataset size (Campos et al., 2016). The density-based clustering method Density Based Spatial Clustering of Applications with Noise (DBSCAN) (Ester et al., 1996) detects anomalies as observations not belonging to any clusters. DBSCAN is advantageous due to its robustness to noise and its ability to identify arbitrarily shaped clusters, but its performance is sensitive to the choice of ε and $MinPts$ (more information on this is provided in Appendix A), and it struggles with varying densities and large datasets due to its quadratic time complexity. To address these issues, Campello et al. (2013) introduced Hierarchical DBSCAN (HDBSCAN), which extends DBSCAN by handling clusters of varying

density, requiring only a single core parameter and providing a more stable cluster hierarchy. Its ability to detect sparse or weakly connected regions makes it suitable for anomaly detection in complex datasets.

Tree-based and kernel-based methods, such as Isolation Forest (iForest) and One Class-Support Vector Machines (OC-SVMs), have also gained prominence in detecting anomalies in time series data. iForest is an unsupervised anomaly detection technique that isolates observations through recursive partitioning: because anomalies are rare and different, they typically require fewer splits to isolate. This method has been applied successfully in fraud detection and other anomaly detection studies applied to financial datasets (Bardet et al., 2019; Saxena et al., 2020). iForest offers several advantages such as, linear time complexity, high scalability, minimal parameter tuning, and no reliance on distributional assumptions. OC-SVM constructs a decision boundary that encloses the majority of observations, treating this region as the “normal” class while flagging external points as anomalies. OC-SVM has been used extensively in fraud detection and financial transaction monitoring (Kavitha et al., 2023). It is robust in high-dimensional spaces, effective with small datasets, and capable of modelling complex non-linear structures via kernel functions (Schölkopf et al., 1999). Its primary limitation is the cubic training complexity, which makes standard OC-SVM unsuitable for large datasets; however, approximate training strategies have been proposed to reduce computational demands while maintaining accuracy (Kang et al., 2019).

Given the limitations of classical methods and the strengths and weaknesses of modern statistical learning approaches, this work adopts an unsupervised, automated ensemble incorporating iForest, OC-SVM, and HDBSCAN. This combination leverages complementary strengths to provide a robust and flexible framework for anomaly detection in time series data.

3 The Dataset and Methodology

3.1 The Dataset

The dataset used in this study is the Securities Holdings Data of Maltese banks which have an ISIN, aggregated over the total bank holdings. It is collected directly from

local banks on a security-by-security basis (based on Regulation ECB/2012/24) and provides information on securities and bonds held by Maltese banks with information on instrument types, holder country and further classifications. The dataset consists of monthly time series representing stocks/positions, defined as the total market value held of each security or bond at the end of the reporting period. Each ISIN is treated as an individual time series. The data covers the period from May 2016 to December 2025 and comprises 7345 time series of varying lengths. For the purposes of this study, all data have been fully anonymized.

3.2 Methodology

This section outlines the methodology employed to identify anomalies in the Granular Securities Holdings Data of Maltese banks. An ensemble modelling approach is adopted, combining the methods discussed in Appendix A to capture different types of anomalies in time series data. The methodology consists of four main stages: data preprocessing, parameter selection, model implementation, and model comparison. All steps were applied independently to each individual time series (ISIN). The implementation was carried out in Python using libraries such as `pandas`, `numpy`, `scikit-learn`, and `hdbscan`.

3.2.1 Data Preprocessing

Before applying iForest, OC-SVM and HDBSCAN to our time series data, the total flow for each period, defined as the difference in market value between two successive months, was computed. This was done since applying anomaly detection techniques on the total flow, rather than the market value makes anomalies easier to detect because it highlights unusual movements while removing underlying trends and level effects. In addition to this, the time series were aligned to a monthly frequency, ensuring a complete sequence of observations. Any missing months within the observed time span were inserted, and corresponding flow values were set to zero to preserve temporal consistency.

3.2.2 HDBSCAN Parameter Selection

The first step in the modelling process involved determining suitable parameters for the HDBSCAN algorithm (see Appendix A.4). Since HDBSCAN is computationally intensive, particularly when applied repeatedly across multiple time series, an empirical parameter selection procedure was performed once and the resulting con-

figuration was used throughout the study. As HDBSCAN operates on feature-based data rather than directly on raw time series, each time series was represented using four summary statistics: the minimum, maximum, mean, and variance. These features were deliberately selected to provide a compact and computationally inexpensive representation of the overall level, range, and variability of each series. Richer temporal or structural descriptors, such as autocorrelation, trend, seasonality, or frequency-domain features, were not included at this stage because the objective was not to construct a comprehensive representation of temporal dynamics, but rather to obtain a simple and consistent feature space for empirical HDBSCAN parameter selection. Limiting the feature set also reduced the dimensionality and computational overhead.

The HDBSCAN algorithm has two key parameters, specifically, `min_cluster_size` and `min_samples`. The parameter `min_cluster_size` defines the smallest grouping considered a cluster, whereas the parameter `min_samples` controls the strictness of the density definition. Larger values of `min_samples` make it more difficult for isolated data points to be included in clusters, thus increasing the number of anomalies identified.

As no established parameter values exist in the literature, `min_samples` and `min_cluster_size` were empirically determined through grid search. The parameter grid consisted of `min_samples` values of 2, 5 and 10, and `min_cluster_size` values of $\lfloor 0.05n \rfloor$, $\lfloor 0.1n \rfloor$, and $\lfloor 0.3n \rfloor$, where n is the number of observations in the time series. These ranges were chosen in such a way as to provide a balanced trade-off between sensitivity to small clusters and robustness to outliers. To ensure algorithmic validity, if the resulting `min_cluster_size` happened to be less than 2, it was automatically set to 2, as smaller values are not permitted. The optimal parameter combination was selected by maximizing the modified Dunn index (Ilc, 2012), which is suitable for evaluating clustering performance in the presence of noise and non-convex clusters. The selected HDBSCAN parameters were then fixed and used consistently across all subsequent experiments.

3.2.3 Isolation Forest and One-Class SVM

The anomaly detection framework combines iForest (see Appendix A.2) and OC-SVM (see Appendix A.3), applied alongside HDBSCAN.

The iForest algorithm was applied directly to the total flow of each period.

iForest relies on two key parameters, namely, `n_estimators` and `contamination`. The former specifies the number of decision trees in the ensemble and was kept at a default value of 100 (scikit-learn Developers, 2025a). The contamination parameter, which controls the expected proportion of anomalies, was varied across three values: 0.05, 0.10, and 0.15.

The OC-SVM algorithm was applied to the sliding window (see Appendix A.1) representation of the data to capture contextual anomalies. The kernel parameter `gamma` was set to the default `scale` (scikit-learn Developers, 2025b), defined as:

$$\gamma = \frac{1}{p \cdot \mathbb{V}ar(X)},$$

where p is the number of features and $\mathbb{V}ar(X)$ is the mean feature variance. The parameter `nu`, representing the upper bound on the fraction of anomalies, was varied across three values: 0.05, 0.10, and 0.15.

The window size for the sliding window transformation was determined independently for each time series using autocorrelation-based period detection. For each ISIN, the autocorrelation function was computed and the lag corresponding to the strongest autocorrelation peak was selected as the estimated period. As a result, the detected period length, and consequently the window size, varied across ISINs rather than being fixed globally across the dataset. This allowed the sliding-window transformation to adapt to the temporal characteristics of each individual time series. In the comparative study by Ermshaus et al. (2022), autocorrelation-based windowing ranked second only to AutoPeriod in terms of anomaly detection performance, while being substantially less computationally intensive. The procedure therefore provided a computationally efficient means of determining a data-dependent window size. Peak stability was not evaluated separately; instead, the strongest autocorrelation peak identified for each time series was used directly as the period estimate.

3.2.4 Model Combinations and Ensemble Approach

To assess the sensitivity of anomaly detection to parameter choices, a total of nine model combinations were constructed using all pairwise combinations of iForest contamination and OC-SVM ν values:

0.05, 0.10, 0.15 $\times$ 0.05, 0.10, 0.15.

For each combination, iForest, OC-SVM, and HDBSCAN were applied, and their outputs were combined using a majority-voting ensemble rule. Specifically, an observation was classified as anomalous if at least two of the three methods identified it as an anomaly:

$$\text{ANOMALY} = \begin{cases} 1 & \text{if } \sum_{m=1}^3 I_m \geq 2, \\ 0 & \text{otherwise,} \end{cases}$$

where I_m is a binary indicator of whether method m flags the observation as anomalous. Majority voting was selected because it provides a simple and interpretable mechanism for combining the decisions of heterogeneous anomaly detection methods without requiring additional weighting parameters or the calibration and normalization of method-specific anomaly scores. This is particularly advantageous in an unsupervised setting, where labelled observations are not available to determine appropriate ensemble weights. Moreover, given that anomaly detection may need to be performed repeatedly across a large number of time series, the low additional computational overhead of majority voting was considered preferable to more complex ensemble strategies. Requiring agreement between at least two methods also limits the influence of anomalies identified by only a single detector, thereby producing a more conservative consensus across the ensemble.

3.2.5 Model Evaluation and Comparison

Given the absence of labelled data, model performance cannot be evaluated using traditional accuracy metrics. Instead, a comparative evaluation framework based on anomaly rates was adopted. For each model combination and each observation date (OBS_DATE), the *total number of observations*, *number of anomalies detected by each method*, *percentage of anomalies detected by each method* and *percentage of final anomalies (ensemble output)* were computed. These results were then summarized in a comparison table displaying, for each model the *average percentage of final anomalies*, *standard deviation of anomaly percentage over time*, *maximum anomaly percentage observed*, and *average anomaly percentages for each individual method*.

In addition, a pivot table was created using OBS_DATE as the index, where each

column represents a model combination and shows the percentage of final anomalies. This makes it easier to compare how the models behave over time and to identify parameter settings that are consistent and reliable. The comparison focuses on selecting models that balance sensitivity with practical usability, favouring those that produce moderate anomaly rates, show limited variation over time, and avoid large spikes.

To provide a simpler statistical benchmark for comparison with the machine-learning approaches, the dataset was also analysed using classical anomaly detection methods. Specifically, the Modified Z-score, IQR, and GESD methods were applied independently to each ISIN.

Prior to applying these methods, an Autoregressive Integrated Moving Average (ARIMA) model was fitted separately to the original series of each ISIN in order to account for its temporal structure. The most appropriate ARIMA specification was selected automatically based on the Akaike Information Criterion (AIC). The resulting residuals were then assessed for stationarity using the Augmented Dickey-Fuller (ADF) test and for normality using the Shapiro-Wilk test. These diagnostics were conducted independently for each ISIN.

The three classical anomaly detection methods were subsequently applied to the ARIMA residuals. The Modified Z-score was based on the median and median absolute deviation, while the IQR method identified observations lying outside the conventional ($1.5 \times \text{IQR}$) bounds. GESD was applied using a significance level of 5%, with the maximum number of candidate outliers restricted to 15% of the observations.

For comparison with the machine-learning results, the detected anomalies were aggregated by observation date. For each classical method, the percentage of anomalies was calculated over time and summarised using the average, standard deviation, minimum, and maximum anomaly percentage.

4 Results and Discussion

This section presents and discusses the results obtained using the anomaly detection framework described in Section 3. As outlined in Section 3, the analy-

sis was conducted using an ensemble approach combining iForest, OC-SVM, and HDBSCAN. Since the dataset is entirely unlabelled, it was not possible to assess the individual methods using conventional supervised performance metrics such as accuracy, precision, or recall. Consequently, the results are evaluated comparatively, focusing on the proportion of anomalies detected, their temporal stability, and the operational burden associated with each parameter combination.

4.1 Computational Performance

The full anomaly detection pipeline was executed separately for each of the nine parameter combinations considered in this study. Table 4.1 reports the computation time for each combination.

Model	Computation Time (seconds)
ISO_0.05.SVM_0.05	845
ISO_0.05.SVM_0.1	820
ISO_0.05.SVM_0.15	865
ISO_0.1.SVM_0.05	985
ISO_0.1.SVM_0.1	1285
ISO_0.1.SVM_0.15	976
ISO_0.15.SVM_0.05	1030
ISO_0.15.SVM_0.1	978
ISO_0.15.SVM_0.15	838

Table 4.1: Computation time for each model.

The results indicate that computation times vary across parameter combinations, ranging from 820 seconds to 1285 seconds. In general, the computational burden is affected by the characteristics of the sliding window representation and the complexity of the fitted models. It is also worth noting that computation time increases approximately linearly with the number of distinct ISINs included in the dataset, since each time series is processed independently.

4.2 Comparison of Parameter Combinations

To evaluate the effect of the different parameter settings, a summary comparison table was constructed. For each model combination, the average percentage of final anomalies, the standard deviation of the final anomaly percentage over time, the maximum anomaly percentage observed, and the average anomaly percentages detected by each individual method were computed. These results are presented in Table 4.2.

Model	Avg Final (%)	Std Final (%)	Max Final (%)	Avg ISO (%)	Avg SVM (%)	Avg HDBSCAN (%)
ISO_0.05_SVM_0.15	8.17	5.80	40.81	6.15	12.57	15.34
ISO_0.05_SVM_0.05	8.21	6.24	39.09	6.15	12.91	15.34
ISO_0.05_SVM_0.10	8.21	6.01	39.86	6.15	12.64	15.34
ISO_0.10_SVM_0.15	8.64	5.81	40.81	10.51	12.57	15.34
ISO_0.10_SVM_0.10	8.69	6.02	39.86	10.51	12.64	15.34
ISO_0.10_SVM_0.05	8.71	6.25	39.09	10.51	12.91	15.34
ISO_0.15_SVM_0.15	9.22	5.83	40.81	15.71	12.57	15.34
ISO_0.15_SVM_0.10	9.27	6.05	39.86	15.71	12.64	15.34
ISO_0.15_SVM_0.05	9.30	6.27	39.09	15.71	12.91	15.34

Table 4.2: Summary of anomaly detection performance across different ISO and SVM parameter combinations.

HDBSCAN parameters were fixed after the empirical parameter selection stage described in Section 3.2.2 and so the average anomaly percentage identified by HDBSCAN remains constant across all model combinations, as seen in Table 4.2. Consequently, differences between the 9 models are driven primarily by the varying parameter values of iForest and OC-SVM.

Table 4.2 shows that the Isolation Forest contamination parameter is the main driver of anomaly detection sensitivity. Increasing the contamination level leads to a consistent increase in the proportion of anomalies detected in the final ensemble output. In contrast, the effect of the One-Class SVM ν parameter is comparatively modest, producing only minor differences within each fixed Isolation Forest setting.

Across all model combinations, the level of variation over time is quite similar, with standard deviation values ranging from about 5.8% to 6.3%. In addition, all models show high peak anomaly percentages, with maximum values exceeding 39% in every case. This indicates that these large spikes in detected anomalies are likely due to patterns in the data itself, rather than being caused by the choice of model parameters alone.

Overall, the combinations with lower Isolation Forest contamination values, particularly `contamination=0.05`, produce a more conservative and manageable number of detected anomalies, while showing similar levels of variation over time compared to the other models. For practical use, these lower-contamination settings may therefore be preferable, especially when the goal is to limit the number of flagged observations that need further investigation.

4.3 Temporal Behaviour of the Final Anomaly Rate

To examine how the model combinations behave over time, the monthly percentage of final anomalies was compared across the nine models. Since the dataset is com-

prised of many time points and usually users only query recent data, only the final 12 months are reported in Table 4.3.

Model	2025-01	2025-02	2025-03	2025-04	2025-05	2025-06	2025-07	2025-08	2025-09	2025-10	2025-11	2025-12
ISO_0.05_SVM_0.05	6.58	9.01	8.08	7.82	10.12	8.81	12.88	19.40	28.19	34.19	35.88	35.79
ISO_0.05_SVM_0.10	6.76	7.73	7.71	8.28	9.94	9.56	12.88	20.56	27.50	31.21	36.38	30.87
ISO_0.05_SVM_0.15	6.76	8.19	7.26	7.18	9.56	10.87	13.36	18.63	26.72	27.83	35.38	29.33
ISO_0.10_SVM_0.05	7.38	9.10	8.98	9.02	10.21	9.18	13.65	19.79	28.39	34.49	35.98	36.21
ISO_0.10_SVM_0.10	7.56	8.01	8.44	9.38	10.03	9.93	13.45	20.85	27.80	31.51	36.58	31.28
ISO_0.10_SVM_0.15	7.47	8.46	7.99	8.65	9.94	11.06	13.84	18.73	26.92	28.13	35.58	29.74
ISO_0.15_SVM_0.05	7.21	9.65	10.53	10.58	10.58	9.37	14.12	20.17	28.59	34.89	36.18	36.92
ISO_0.15_SVM_0.10	7.21	8.64	10.07	10.76	10.40	10.22	13.74	21.43	28.09	31.81	36.88	31.79
ISO_0.15_SVM_0.15	7.21	9.01	9.62	10.12	10.21	11.34	14.12	19.31	27.21	28.33	35.88	30.15

Table 4.3: Monthly percentage of final anomalies detected using Machine Learning Method for the period January 2025-December 2025.

Table 4.3 reports the monthly percentage of final anomalies for the last 12 months of the sample. This period is particularly informative because it contains the highest anomaly rates and the clearest divergence between parameter combinations.

A pronounced upward trend is visible from January 2025 to November 2025 across all nine model combinations. This indicates a substantial increase in the proportion of observations exhibiting patterns that are identified as anomalous by the models. However, in the absence of labelled data, the underlying cause and validity of these detected anomalies cannot be established from the model outputs alone and require further investigation. In practice, observations identified as anomalous would subsequently be reviewed to determine whether they reflect genuinely unusual behaviour and, where appropriate, investigated further, for example by querying the relevant Reporting Agent. Importantly, the same broad increase is observed across all nine model configurations. This does not constitute empirical evidence of robustness, but it indicates that the observed pattern is consistent across the parameter configurations considered and is not attributable solely to a single parameter choice.

The differences between model combinations become more evident as anomaly intensity increases. During the earlier months of 2025, the spread between the most conservative and most aggressive models is relatively small. However, by October to December 2025 this gap widens substantially. For example, in December 2025 the lowest anomaly percentage is 29.33% for ISO_0.05_SVM_0.15, while the highest is 36.92% for ISO_0.15_SVM_0.05. This indicates that parameter choice has a greater effect during periods of heightened abnormality.

Consistent with the results in Table 4.2, the Isolation Forest contamination parameter remains the main source of variation between models. Combinations involving ISO = 0.15 generally produce higher anomaly percentages, whereas com-

binations with $\text{ISO} = 0.05$ remain more conservative. By contrast, varying the One-Class SVM ν parameter produces smaller within-group differences, indicating that its impact on the final ensemble output is more limited.

From an operational perspective, the lower-contamination models appear preferable, since they preserve the same broad anomaly pattern while generating a smaller proportion of flagged observations. This is particularly important when the objective is to balance anomaly sensitivity against the burden of subsequent querying and investigation.

4.4 Comparison with Classical Methods

To further assess the anomaly patterns identified by the machine-learning approaches, their results were compared with those obtained using the classical anomaly detection methods introduced previously, namely the Modified Z-score, IQR, and GESD. This comparison provides a simpler statistical benchmark and allows the extent to which the detected anomaly patterns differ between classical and machine-learning-based approaches to be examined.

However, the results of the classical methods should be interpreted with caution. Prior to anomaly detection, the residuals obtained from the fitted ARIMA models were tested separately for each ISIN for stationarity using the ADF test and for normality using the Shapiro-Wilk test. These diagnostics showed that most of the time series did not satisfy the required data assumptions, particularly with respect to normality and stationarity. Consequently, the assumptions underlying some of the classical methods may not hold adequately for a substantial proportion of the dataset, and the resulting anomaly classifications may therefore be less reliable or potentially misleading. The classical results are thus primarily considered as benchmark comparisons rather than as definitive classifications of anomalous observations.

Table 4.4 summarises the anomaly rates obtained from the three classical anomaly detection methods, while Table 4.2 reports the corresponding results for the machine-learning ensemble configurations.

Method	Avg (%)	Std (%)	Max (%)
Modified Z-score	11.88	8.59	58.06
IQR	13.84	9.68	62.96
GESD	7.65	6.48	36.05

Table 4.4: Summary of anomaly detection performance across different classical methods.

Comparing Tables 4.2 and 4.4, it can be seen that the ensemble models produced relatively moderate average anomaly rates, ranging from 8.17% to 9.30% across the nine parameter combinations. These values are lower than the average anomaly rates obtained using the Modified Z-score and IQR methods, which were 11.88% and 13.84%, respectively. GESD was the most conservative classical method, with an average anomaly rate of 7.65%, which is slightly below the range observed for the machine-learning ensembles.

The variability of the detected anomaly rates was also broadly comparable. The standard deviation of the machine-learning ensemble anomaly percentages ranged from 5.80% to 6.27%, compared with 8.59% for the Modified Z-score, 9.68% for IQR, and 6.48% for GESD. The classical Modified Z-score and IQR methods therefore exhibited greater variation in anomaly rates over time, while GESD showed variability closer to that of the machine-learning ensembles. A similar difference is observed in the maximum anomaly rates: the machine-learning ensembles reached maxima of approximately 39 – 41%, whereas the Modified Z-score and IQR reached substantially higher maxima of 58.06% and 62.96%, respectively. GESD again produced the lowest maximum anomaly rate, at 36.05%.

From an operational perspective, these differences have direct implications for the number of observations requiring manual review and possible follow-up with Reporting Agents. Since each detected anomaly may need to be assessed before deciding whether further clarification is required, higher anomaly rates translate into a greater investigation and querying burden. On average, GESD would generate the lowest number of cases for review, followed by the machine-learning ensemble configurations. The Modified Z-score and, particularly, the IQR method would generate a larger number of alerts and therefore a greater operational burden. The machine-learning ensemble thus provides a comparatively moderate alert rate while avoiding the substantially higher anomaly frequencies observed for some of the classical methods.

However, a lower anomaly rate should not in itself be interpreted as evidence

of superior detection performance. In the absence of labelled data, it is not possible to determine whether a lower number of alerts reflects more selective identification of meaningful anomalies or reduced sensitivity to genuinely unusual observations. The operational comparison therefore highlights a trade-off between the number of cases requiring investigation and the sensitivity of the anomaly detection approach. In addition, since the stationarity and normality diagnostics were not satisfied for many ISINs, the results of the classical methods should be interpreted cautiously when used as benchmarks against the machine-learning approaches.

Table 4.5 shows the monthly percentage of anomalies detected using each classical method for the period January 2025–December 2025. Recall that the corresponding results table for the machine learning models are found in Table 4.3.

Method	2025-01	2025-02	2025-03	2025-04	2025-05	2025-06	2025-07	2025-08	2025-09	2025-10	2025-11	2025-12
Modified Z-Score	7.83	7.02	9.34	7.81	6.88	4.44	4.79	2.84	3.14	3.40	3.16	3.19
IQR	8.46	7.83	10.69	10.11	7.81	5.00	5.45	3.22	3.04	4.44	3.25	3.94
GESD	4.50	2.88	3.41	3.77	3.72	2.27	3.38	1.42	1.62	2.08	1.44	2.16

Table 4.5: Monthly percentage of anomalies detected using each Classical Method for the period January 2025–December 2025.

The monthly comparison in Tables 4.3 and 4.5 reveals a marked difference in the temporal behaviour of the machine-learning and classical approaches. During the first half of 2025, the anomaly percentages produced by the machine-learning ensemble configurations remained relatively moderate, generally ranging between approximately 6% and 11%. Over the same period, the Modified Z-score and IQR methods produced broadly similar or slightly lower anomaly rates, while GESD was consistently more conservative.

From July 2025 onwards, however, the two approaches diverged considerably. The machine-learning ensembles identified an increasing proportion of observations as anomalous, with anomaly rates rising from approximately 13 – 14% in July to around 18–21% in August, approximately 27–29% in September, and above 30% for most configurations in October and November. By contrast, the classical methods did not exhibit the same upward pattern. The Modified Z-score and IQR anomaly rates generally declined during the second half of the year and remained mostly between approximately 3% and 5%, while GESD remained lower still, generally around 1 – 3%.

This difference suggests that the increase detected by the machine-learning approaches in the second half of 2025 is not reproduced by the classical statisti-

cal methods. Rather than providing confirmation of the machine-learning results, the classical benchmarks therefore highlight a substantial difference in how the two methodological approaches respond to the underlying data. One possible explanation is that the machine-learning methods are able to capture changes in the structure or multivariate characteristics of the observations that are not identified by classical residual-based thresholding methods. However, in the absence of labelled data, it is not possible to determine whether the higher machine-learning anomaly rates reflect greater sensitivity to genuinely unusual behaviour or whether they represent over-identification of anomalous observations.

The comparison also has practical implications for the subsequent review process. In the second half of 2025, the classical approaches would have generated substantially fewer cases requiring manual assessment and possible follow-up with Reporting Agents, whereas the machine-learning ensembles would have produced a considerably larger investigation workload. Nevertheless, this lower operational burden should not be interpreted as evidence that the classical methods are preferable, since fewer alerts may also reflect lower sensitivity to meaningful changes in the data. Moreover, as the stationarity and normality diagnostics were not satisfied for many ISINs, the classical anomaly rates themselves should be interpreted cautiously. Overall, the divergence between the two approaches reinforces the need for subsequent case-level investigation to determine which detected observations represent genuine anomalous behaviour.

4.5 Discussion

Taken together, the results indicate that the machine-learning ensemble provides an interpretable and flexible framework for identifying unusual movements in securities holdings data. In the absence of labelled observations, however, the detected anomalies cannot be formally validated against ground truth. The results should therefore be interpreted as patterns that are consistent across the parameter configurations considered, rather than as evidence of robustness or predictive superiority. In particular, the broad increase in anomaly rates observed during the second half of 2025 was identified across all nine machine-learning configurations, suggesting that this pattern was not driven solely by a particular parameter choice.

The comparison with classical anomaly detection methods provides an additional perspective. The Modified Z-score, IQR, and GESD methods offer compu-

tationally simple benchmarks, but their application requires greater reliance on the statistical properties of the underlying data. In the present analysis, the ARIMA residuals were tested for stationarity and normality separately for each ISIN, and these assumptions were not satisfied for a substantial proportion of the series. Consequently, the results obtained from the classical methods should be interpreted cautiously. By contrast, the machine-learning methods are more data-adaptive and do not require normality or stationarity to hold in the same manner, making them potentially more suitable for heterogeneous financial time series.

The two methodological approaches also produced different temporal anomaly patterns. While the machine-learning ensembles displayed a pronounced increase in anomaly rates during the second half of 2025, the classical methods did not reproduce the same pattern and generally produced lower anomaly rates over this period. This divergence suggests that the methods respond differently to changes in the underlying data. The greater flexibility of the machine-learning models may allow them to identify patterns that are not captured by residual-based thresholding approaches. However, without labelled anomalies, it cannot be determined whether the additional detections represent genuinely unusual behaviour or increased model sensitivity.

From a computational perspective, the classical methods did not provide a substantial advantage once the complete procedure was considered. Although the Modified Z-score, IQR, and GESD rules themselves are computationally inexpensive, their application required fitting an ARIMA model separately to each ISIN and performing residual diagnostic tests. The complete classical analysis required approximately 900 seconds, while the machine-learning configurations required between 820 and 1,285 seconds. Thus, when the necessary preprocessing is included, the overall computational requirements of the two approaches were broadly comparable.

The machine-learning framework nevertheless requires an initial parameter-selection stage. In particular, the Isolation Forest contamination parameter and One-Class Support Vector Machine (SVM) ν influence the number of observations classified as anomalous, while HDBSCAN also requires appropriate parameter selection. This represents an additional implementation requirement compared with simpler threshold-based methods. However, once suitable parameter settings have been identified through initial experimentation, the same configuration can be ap-

plied consistently in subsequent executions, provided that the characteristics of the underlying data do not change substantially.

The choice of parameter values also has important operational implications. Higher contamination and ν values generally increase the number of detected anomalies and therefore the number of cases requiring subsequent review. Lower parameter values provide a more conservative set of anomaly flags while preserving the same broad temporal patterns observed across the alternative configurations. Since each detected anomaly may require manual assessment and, where necessary, further clarification from the relevant Reporting Agent, limiting unnecessary alerts is an important practical consideration. Among the configurations considered, the lower-parameter combinations therefore appear to offer a more appropriate balance between sensitivity and operational workload.

Overall, the machine-learning ensemble appears to provide the more suitable framework for the characteristics and intended application of the present dataset, particularly when more conservative parameter settings are used. This conclusion should be interpreted in terms of practical suitability rather than demonstrated superiority in detection accuracy. The anomaly detection framework should ultimately be viewed as a screening mechanism: observations identified as anomalous require subsequent case-level investigation to determine whether they represent genuine unusual behaviour, data-quality issues, or explainable changes in reporting activity.

5 Concluding Remarks

The objective of this study was to develop a framework for detecting anomalies in the Granular Securities Holding Data of Maltese banks. Given the absence of labelled data and the heterogeneous nature of the dataset, an ensemble approach combining Isolation Forest, One-Class SVM, and HDBSCAN was adopted. The parameters of HDBSCAN were determined empirically and subsequently fixed, while multiple combinations of the Isolation Forest contamination parameter and the One-Class SVM ν parameter were examined to assess their influence on the resulting anomaly patterns.

The findings indicate that the Isolation Forest contamination parameter has

the strongest influence on the overall proportion of detected anomalies, with higher values generally producing a larger number of anomaly flags. Changes in the One-Class SVM ν parameter had a comparatively smaller effect on the final ensemble output. Importantly, the same broad temporal pattern was observed across the nine parameter configurations, particularly the increase in anomaly rates during the second half of 2025. In the absence of labelled observations, this should not be interpreted as evidence of model robustness or detection accuracy, but rather as an indication that the observed pattern is consistent across the parameter settings considered.

The machine-learning results were also compared with simpler classical anomaly detection methods, namely the Modified Z-score, IQR, and GESD. These approaches provided a useful statistical benchmark, although their application required fitting an ARIMA model separately to each ISIN and assessing the resulting residuals for stationarity and normality. Since these assumptions were not satisfied for a substantial proportion of the series, the classical results should be interpreted cautiously. The classical methods also produced different temporal anomaly patterns from the machine-learning ensemble, particularly during the second half of 2025, further illustrating that the two approaches respond differently to the underlying data.

From a practical perspective, the lower-parameter machine-learning configurations appear to offer the most suitable balance between sensitivity and operational usability. In particular, configurations with lower Isolation Forest contamination values produced a more conservative number of anomaly flags while preserving the same broad temporal structure observed under higher parameter settings. This is important in an operational environment, since each flagged observation may require manual assessment and, where necessary, further clarification from the relevant Reporting Agent. Lower alert volumes can therefore reduce the downstream investigation burden without materially changing the overall anomaly pattern identified by the ensemble.

The machine-learning approach also appears well suited to the characteristics of the dataset because it is more flexible and less dependent on strict distributional assumptions than the classical alternatives. Although the machine-learning methods require an initial hyperparameter-selection stage, once suitable configurations have been established they can be applied consistently in subsequent executions,

provided that the characteristics of the underlying data remain broadly comparable. Moreover, when the preprocessing required for the classical methods is taken into account, the overall computational cost of the two approaches was found to be broadly similar.

Several opportunities for further research remain. More extensive hyperparameter optimisation could be explored to assess whether alternative configurations improve the practical behaviour of the ensemble. Future work could also investigate alternative ensemble strategies, including weighted voting, anomaly-score aggregation, or rank-based fusion, as well as a broader set of model architectures. In addition, further validation could be undertaken through synthetic anomaly injection, stability analysis, or expert assessment. Over time, the framework could also evolve towards a semi-supervised setting as detected anomalies are reviewed and labelled through the investigation process. Finally, the proposed framework is not limited to securities holdings data and could potentially be adapted to other financial applications involving large, heterogeneous, and unlabelled time series datasets.

Appendix A

Statistical Theory

A.1 Sliding Window Approach

Time series data are sequential, and most machine learning methods cannot directly handle raw sequences of arbitrary length. As a result, a sliding window approach may be used to transform the time series into a set of fixed length subsequences. Given a univariate time series $\mathbf{x} = \{x_1, \dots, x_T\}$ and a window of size w , vectors $\tilde{\mathbf{x}}_t = (x_t, x_{t+1}, \dots, x_{t+w-1}) \in \mathbb{R}^w$ are constructed $\forall t \in \{1, \dots, T - w + 1\}$ to form an independent and identically distributed embedding data set $\tilde{\mathbb{X}}$:

$$\tilde{\mathbb{X}} = \begin{pmatrix} \tilde{\mathbf{x}}_1 \\ \tilde{\mathbf{x}}_2 \\ \vdots \\ \tilde{\mathbf{x}}_{T-w+1} \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & \dots & x_w \\ x_2 & x_3 & \dots & x_{w+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{T-w+1} & x_{T-w+2} & \dots & x_T \end{pmatrix} \quad (\text{A.1})$$

where each row $\tilde{\mathbf{x}}_t \in \mathbb{R}^w$ is a sample corresponding to a snapshot of the time series over a short time window w and each column is an attribute (feature).

A.2 Isolation Forest

As may be recalled, iForest (Liu et al., 2008) is an unsupervised anomaly detection method based on the idea of isolating individual observations from the rest of the data. Since anomalies are rare and different, they are more easily isolated than normal points. iForest uses recursive partitioning to separate the data until all points are isolated. Those data points requiring fewer splits are then identified as anomalies.

The iForest algorithm begins by constructing t isolation trees (iTrees) (see Definition 1) to form a forest.

Definition 1 Isolation Tree

An isolation tree is a proper binary tree in which each node v is either:

- *an external node (leaf) with no children;*
- *an internal node containing a test and two child nodes v_l and v_r .*

At an internal node, in a univariate setting, a test consists of a split values τ that divides the data points into v_l and v_r .

Consider a sample of data $X = \{x_1, \dots, x_T\}$, an iTree is constructed in the following way:

1. A split value τ is chosen uniformly at random between the minimum and maximum value of the sample:

$$\tau \sim U(\min(X), \max(X)).$$

2. Using the split value τ , a cutting plane is generated to partition the data in two as follows:
 - (a) Observations, x_l , whose values are less than τ , i.e. $x < \tau$, $x \in X$, are assigned to the left of the split point forming a left subtree sample space;
 - (b) Observations, x_r , whose values are greater than or equal to τ , i.e. $x \geq \tau$, $x \in X$, are assigned to the right of the split point forming a right subtree sample space.
3. Steps 1 and 2 are recursively repeated in the left and right subspaces until:
 - (a) The tree reaches a height limit;
 - (b) The subspace contains only one data point;
 - (c) All data in X have the same values.

After constructing t random iTrees, for each observation x , the number of splits required to isolate it within each tree is recorded as the path length, denoted by $h(x)$. Since an iTree has the same recursive partitioning structure as a binary search tree (BST) (Liu et al., 2008), the expected path length of an unsuccessful search in a BST can be used as an approximation. This expectation is given by:

$$c(T) = 2H(T-1) - \frac{2(T-1)}{T}, \tag{A.2}$$

where $H(T) = \ln(T) + 0.5772$ (Qin & Lou, 2019). Finally, the anomaly score for each observation is computed as:

$$s(x) = 2^{\frac{-\mathbb{E}(h(x))}{c(T)}}, \quad (\text{A.3})$$

where $\mathbb{E}(h(x)) = \frac{1}{t} \sum_{l=1}^t h_l(x)$, $\forall x \in X$. Anomaly scores close to 1 suggest that the data point was isolated with very few splits and thus, is an anomaly. Conversely, scores close to 0 indicate that the observation required many splits to isolate and thus, can be considered as a regular observation. Anomaly scores around 0.5 are indicative of uncertainty and may require further analysis (Liu et al., 2008).

A.3 One Class Support Vector Machines

SVMs (Cortes & Vapnik, 1995) are powerful algorithms typically employed in supervised learning tasks, such as classification and regression. In principle, the task of anomaly detection can also be framed as a binary classification problem in which one class represents normal observations and the other class represents anomalies. Nevertheless, in practice, class labels are often unavailable. Thus, OC-SVMs (Schölkopf et al., 1999) were proposed as a special case of SVMs, specifically designed for unsupervised anomaly detection.

The goal of the OC-SVM algorithm is to identify the best possible decision boundary, referred to as a *hyperplane*, that encloses the most data, treating this region as the “normal” class, with data points falling outside this boundary being flagged as anomalies. The hyperplane exists within a p -dimensional space, where p is the number of attributes within the data. The optimal hyperplane is the one that maximizes the *margin*, i.e. the distance between the closest data points of each class.

When the data is linearly separable, the OC-SVM algorithm can directly compute the hyperplane that separates the two classes. However, when the data is not linearly separable, the algorithm can still effectively classify the data by implementing a technique called the *kernel trick*. The general idea of the kernel trick is to map the data into a higher-dimensional space where it might become linearly separable, allowing the algorithm to find an optimal hyperplane.

The margin and decision boundary are influenced by the *support vectors*. The support vectors are the data points which are closest to the hyperplane, essentially

defining its position. If these support vectors were removed, the hyperplane would shift, potentially altering the classification of other observations. Having loosely explained the OC-SVM algorithm, the mathematical formulation will now be presented.

Given data $\mathbf{x} = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$, where $\mathbf{x}_t$ is a p -dimensional row vector $t = 1, \dots, T$, the p -dimensional hyperplane that separates the data from the origin with maximum margin given by:

$$\mathbf{w} \cdot \mathbf{x} = b, \quad (\text{A.4})$$

where $\mathbf{w}$ is a p -dimension column vector of weights orthogonal to the hyperplane and b is the offset from the origin. Ideally, most points satisfy $\mathbf{w} \cdot \mathbf{x} \geq b$ (on the “normal side” of the hyperplane), and the margin $\frac{b}{\|\mathbf{w}\|}$ should be large. To maximize the margin, the following objective function is minimized:

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 - b \quad (\text{A.5})$$

$$\text{s.t. } \mathbf{w} \cdot \mathbf{x}_t \geq b, \quad \forall t. \quad (\text{A.6})$$

In practice, real world data is rarely perfectly separable, and so, to address this, the algorithm introduces slack variables ξ_t , $t = 1, \dots, T$, to allow for some misclassifications. The slack variables ξ_t measure the degree of misclassification for each data point, enabling the algorithm to handle situations where some data points are on the wrong side of the decision boundary. Thus, in order to determine the hyperplane, the following objective function is minimized instead:

$$\min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{\nu T} \sum_{t=1}^T (\xi_t) - b \quad (\text{A.7})$$

$$\text{s.t. } \mathbf{w} \cdot \mathbf{x}_t \geq b - \xi_t, \quad \xi_t \geq 0, \quad t = 1, \dots, T, \quad (\text{A.8})$$

where $\nu \in (0, 1]$ is a hyperparameter that controls the influence of the slack variable. Equation (A.7) is referred to as the *primal problem* formulation for OC-SVMs. To simplify the optimization process, instead of minimizing the primal problem in

Equation (A.7), its dual problem is maximized:

$$\max_{\alpha} -\frac{1}{2} \sum_{t,k=1}^T \alpha_t \alpha_k (\mathbf{x}_t^{\top} \mathbf{x}_k) \quad (\text{A.9})$$

$$\text{s.t.} \quad \sum_{t=1}^T \alpha_t = 1, \quad 0 \leq \alpha_t \leq \frac{1}{\nu T}, \quad \forall t, \quad (\text{A.10})$$

where α_t is a Lagrangian multiplier. As evident from Equation (A.9), the dual formulation is advantageous, as it involves only dot products between data points, facilitating computational efficiency.

Unfortunately, there are instances where the data is not linearly separable in the original feature space $\mathbb{R}^p$, and so, the kernel trick is implemented. The kernel trick involves using a kernel function $K(\cdot)$ to map the data to a higher dimensional space, $\mathbb{R}^{p'}$, $p' > p$, where a linear decision boundary can be found. Thus, Equation (A.9) becomes:

$$\max_{\alpha} -\frac{1}{2} \sum_{t,k=1}^T \alpha_t \alpha_k K(\mathbf{x}_t^{\top} \mathbf{x}_k). \quad (\text{A.11})$$

Common kernel functions include the linear kernel, the polynomial kernel and the radial basis function (RBF) kernel. In the application portion, the RBF kernel, given by $K(\mathbf{x}_t, \mathbf{x}_k) = \exp(-\gamma \|\mathbf{x}_t - \mathbf{x}_k\|^2)$, is used since it can be used for all kinds of datasets, i.e. small or large with both high and low dimension (Yokkampon et al., 2021). After fitting the model, the OC-SVM identifies any anomalous points by computing the decision function $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} - b$. If $f(\mathbf{x}) > 0$, the observation is deemed normal, otherwise it is anomalous.

A.4 HDBSCAN

HDBSCAN (Campello et al., 2013) extends the density-based clustering framework introduced by DBSCAN (Ester et al., 1996) by incorporating a hierarchical representation of clusters across varying density levels. This hierarchical approach allows the identification of clusters with different densities and automatically distinguishes points that do not belong to any dense region, which are labelled as outliers. Thus, even though it is primarily utilized as a clustering method, it can also serve as an outlier detection method.

Consider a dataset containing n observations, each characterized by p attributes.

This dataset can be represented as an $(n \times p)$ -data matrix $\mathbb{X}$:

$$\mathbb{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} \quad (\text{A.12})$$

The data matrix $\mathbb{X}$ can also be represented as $\mathbb{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix}$, where $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^\top$,

$i = 1, \dots, n$, is a p -dimensional vector of observations on all attributes for the i^{th} observation. Let $\mathbb{D}$ be an $(n \times n)$ -matrix comprising the pairwise distances $d(\mathbf{x}_i, \mathbf{x}_r)$, $\mathbf{x}_i, \mathbf{x}_r \in \mathbb{X}, i, r = 1, \dots, n, i \neq r$, for a suitable distance metric $d(\cdot, \cdot)$. In DBSCAN, the density around a data point $\mathbf{x}_i \in \mathbb{R}^p$, is measured by defining a region around said point, and examining the number of data points within the designated region. These data points are referred to as neighbours. The size of the region is determined by a hyperparameter denoted as ε , which represents the maximum distance between two points for them to be considered as neighbours. In addition to ε , the DBSCAN algorithm uses another hyperparameter *MinPts* which is a criterion which specifies the minimum number of data points required for a region to be considered “dense”. Density-based clusters are defined based on core points alone:

Definition 2 Core Point

*An observation $\mathbf{x}_r$ is said to be a core point, with respect to ε and *MinPts*, if it has a minimum number of other observations (specified by *MinPts*) within a given ε of itself, i.e., if $|\mathcal{N}_\varepsilon(\mathbf{x}_r)| \geq \text{MinPts}$, where $\mathcal{N}_\varepsilon(\mathbf{x}_r) = \{\mathbf{x}_i \in \mathbb{X} | d(\mathbf{x}_i, \mathbf{x}_r) \leq \varepsilon\}$ is the ε -neighbourhood of $\mathbf{x}_r$ and $|\cdot|$ denotes cardinality. If a data point is not a core point, it is considered to be an outlier.*

Definition 3 ε -Reachability

*Two core points $\mathbf{x}_r$ and $\mathbf{x}_q$, $r, q = 1, \dots, n, r \neq q$, are said to be ε -reachable with respect to ε and *MinPts*, if $\mathbf{x}_r \in \mathcal{N}_\varepsilon(\mathbf{x}_q)$ and $\mathbf{x}_q \in \mathcal{N}_\varepsilon(\mathbf{x}_r)$.*

Definition 4 Density Connectedness

*Two core points $\mathbf{x}_r$ and $\mathbf{x}_q$ are said to be density-connected, with respect to ε and *MinPts*, if there are directly or transitively ε -reachable.*

Definition 5 Cluster

A cluster C , with respect to ε and $MinPts$, is a non-empty maximal subset of $\mathbb{X}$ such that each pair of observations in C is density-connected.

Now, based on the definitions of *core point*, ε -reachability, *density-connectedness* and *cluster*, (see Definitions 2-5), an algorithm DBSCAN* can be devised, whereby clusters are identified as components of a graph in which the elements in $\mathbb{X}$ are vertices and each pair of vertices is adjacent if and only if the corresponding data points are ε -reachable with respect to user defined parameters ε and $MinPts$. Non-core points are considered to be outliers.

HDBSCAN has a single-input parameter $MinPts$. Different density-levels in the generated density-based cluster hierarchy will then correspond to different values of ε . In order to properly formulate the density-based cluster hierarchy with respect to a value of $MinPts$, the concepts of *core-distance*, ε -core point, *mutual reachability distance* and *mutual reachability graph* will be defined.

Definition 6 Core Distance

The core distance of a data point $\mathbf{x}_r \in \mathbb{X}$ with respect to $MinPts$, $d_{core}(\mathbf{x}_r)$, is the distance from $\mathbf{x}_r$ to its $MinPts$ -nearest neighbour including $\mathbf{x}_r$.

Definition 7 ε -Core Point

A data point $\mathbf{x}_r \in \mathbb{X}$ is called an ε -core point if $d_{core}(\mathbf{x}_r) \leq \varepsilon$, $\forall \varepsilon$, with respect to $MinPts$.

Definition 8 Mutual Reachability Distance

The mutual reachability distance between two observations $\mathbf{x}_r$ and $\mathbf{x}_q$ in $\mathbb{X}$, with respect to $MinPts$, is defined as $d_{mReach}(\mathbf{x}_r, \mathbf{x}_q) = \max\{d_{core}(\mathbf{x}_r, d_{core}(\mathbf{x}_q)), d(\mathbf{x}_r, \mathbf{x}_q)\}$.

Definition 9 Mutual Reachability Graph

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined as a structure that consists of a set of entities, called nodes (or vertices) $\mathcal{V}$, and a set of relationships called edges $\mathcal{E}$, between these nodes. A mutual reachability graph, G_{MinPts} , is a complete graph in which the observations of $\mathbb{X}$ are vertices and the weights of each edge is the mutual reachability distance,

with respect to $MinPts$, between the respective pair of observations.

HDBSCAN detects clusters of varying density and automatically identifies outliers by constructing a hierarchy of clusters based on the mutual reachability distance. The algorithm first computes the core distance for each point and constructs the mutual reachability graph. A minimum spanning tree of this graph is then generated, and clusters are detected by progressively removing edges in decreasing order of mutual reachability distance. Stable clusters are extracted from the resulting hierarchy, while data points not belonging to any stable cluster are considered to be outliers. To improve interpretability, Campello et al. (2013) proposed a hierarchy simplification procedure that condenses the clusters that remain well-defined across significant density intervals, creating a compact representation of the most stable and meaningful clusters.

References

- Bardet, J., Lang, G., Mameri, A., & Ould Said, E. H. (2019). Outlier detection for temporal data using random forests: Application to financial datasets. In *International workshop on machine learning and data mining in pattern recognition* (pp. 253–267). Springer, Cham. doi: 10.1007/978-3-030-22999-3_20
- Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). Lof: identifying density-based local outliers. In *Proceedings of the 2000 acm sigmod international conference on management of data* (p. 93–104). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/342009.335388> doi: 10.1145/342009.335388
- Campello, R. J. G. B., Moulavi, D., & Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In J. Pei, V. S. Tseng, L. Cao, H. Motoda, & G. Xu (Eds.), *Advances in knowledge discovery and data mining* (pp. 160–172). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Campos, G. O., Zimek, A., Sander, J., Campello, R. J. G. B., Micenková, B., Schubert, E., ... Houle, M. E. (2016). On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, 30(4), 891–927. doi: 10.1007/s10618-015-0444-8
- Chandola, V., Banerjee, A., & Kumar, V. (2009, July). Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3).
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297.
- Ermschaus, A., Schäfer, P., & Leser, U. (2022). Window size selection in unsupervised time series analytics: A review and benchmark. In *Advanced analytics and learning on temporal data: 7th ecml pkdd workshop, aaltd 2022, grenoble, france, september 19–23, 2022, revised selected papers* (p. 83–101). Berlin, Heidelberg: Springer-Verlag. Retrieved from https://doi.org/10.1007/978-3-031-24378-3_6 doi: 10.1007/978-3-031-24378-3_6
- Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Knowledge discovery and data mining*. Retrieved from <https://api.semanticscholar.org/CorpusID:355163>
- Ilc, N. (2012). Modified dunn’s cluster validity index based on graph theory. *Przegląd Elektrotechniczny*, 88(2), 126–131. (In English)
- Kang, S., Kim, D., & Cho, S. (2019). Approximate training of one-class support vector machines using expected margin. *Computers and Industrial Engineering*, 130, 772–778. Retrieved from

- <https://www.sciencedirect.com/science/article/pii/S0360835219301652> doi: <https://doi.org/10.1016/j.cie.2019.03.029>
- Kavitha, N., Vineetha, K., Naga Lakshmi, K., Divya Lashmi Deve, L., & Kranthi Kumar, K. (2023). An effective one-class support vector machine based fraud detection in financial transactions. *International Journal of Creative Research Thoughts*, 11(11). (IJCRT2311033)
- Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In *2008 eighth ieee international conference on data mining* (p. 413-422).
- Neloy, A. A., & Turgeon, M. (2024). A comprehensive study of auto-encoders for anomaly detection: Efficiency and trade-offs. *Machine Learning with Applications*, 17, 100572. Retrieved from <https://www.sciencedirect.com/science/article/pii/S2666827024000483> doi: <https://doi.org/10.1016/j.mlwa.2024.100572>
- Qin, Y., & Lou, Y. (2019). Hydrological time series anomaly pattern detection based on isolation forest. In *2019 ieee 3rd information technology, networking, electronic and automation control conference (itnec)* (p. 1706-1710).
- Rosner, B. (1983). Percentage points for a generalized esd many-outlier procedure. *Technometrics*, 25(2), 165–172. Retrieved from <https://doi.org/10.1080/00401706.1983.10487848> doi: 10.1080/00401706.1983.10487848
- Saxena, R., Prasad, M., Gupta, K., & Bharadwaj, K. K. (2020). Anomaly detection in financial transactions using machine learning and deep learning approaches. In *International conference on innovative computing and communications* (pp. 231–242). Springer, Singapore. doi: 10.1007/978-981-15-5113-0_22
- Schölkopf, B., Williamson, R. C., Smola, A., Shawe-Taylor, J., & Platt, J. (1999). Support vector method for novelty detection. In S. Solla, T. Leen, & K. Müller (Eds.), *Advances in neural information processing systems*. MIT Press.
- scikit-learn Developers. (2025a). sklearn.ensemble.isolationforest — scikit-learn 1.7.2 documentation [Computer software manual]. Retrieved from <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html>
- scikit-learn Developers. (2025b). sklearn.svm.oneclasssvm — scikit-learn documentation [Computer software manual]. Retrieved from <https://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html>
- Shakil, A., Khalighi, M. A., Pudlo, P., Leclerc, C., Laplace, D., Hamon, F., & Boudonne, A. (2023). Outlier detection in non-stationary time series applied to sewer network monitoring. *Internet of Things*, 21, 100654. Retrieved from <https://www.sciencedirect.com/science/article/pii/S2542660522001354> doi: <https://doi.org/10.1016/j.iot.2022.100654>
- Wilcox, R. R. (2002). 3—summarizing data. In *Applying contemporary statistical techniques* (1st ed., pp. 55–91). Burlington: Academic Press.

- Yang, J., & Liu, L. (2025, 06). Robust anomaly detection in financial markets using lstm autoencoders and generative adversarial networks.
doi: 10.20944/preprints202506.1907.v1
- Yaro, A. S., Maly, F., & Prazak, P. (2023). Outlier detection in time-series receive signal strength observation using z-score method with s_n scale estimator for indoor localization. *Applied Sciences*, 13(6). Retrieved from <https://www.mdpi.com/2076-3417/13/6/3900>
- Yaro, A. S., Maly, F., Prazak, P., & Mały, K. (2024). Outlier detection performance of a modified z-score method in time-series rss observation with hybrid scale estimators. *IEEE Access*, 12, 12785-12796.
- Yokkampon, U., Chumkamon, S., Mowshowitz, A., Fujisawa, R., & Hayashi, E. (2021, June). Anomaly detection using support vector machines for time series data. *Engineering Journal*, 8(1), 41–46.