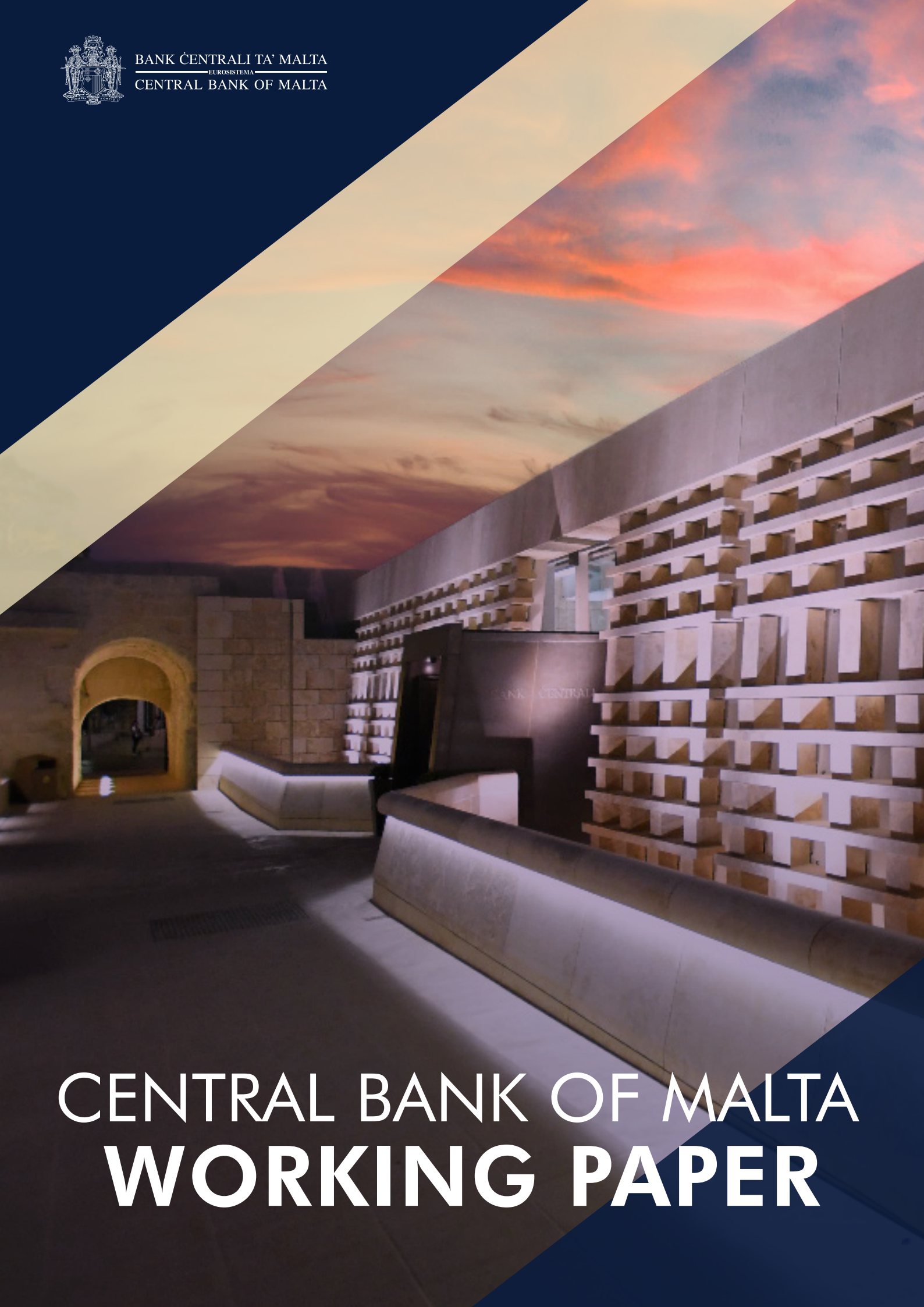

BANK ĊENTRALI TA' MALTA
EUROSISTEMA
CENTRAL BANK OF MALTA

CENTRAL BANK OF MALTA
WORKING PAPER

BANK ĊENTRALI TA' MALTA
EUROSISTEMA
CENTRAL BANK OF MALTA

Detecting Outliers in Malta Pension Schemes and Insurance Corporations Datasets:

A Machine Learning Approach[1]

Sarah Axiaq[2] and Kristen Carabott[3]

WP/06/2024

[1] We would like to thank Mr François Coppens for reviewing an earlier version of this paper and suggesting areas of improvement. We would also like to thank Mr Jesmond Pule and Ms Karen Caruana for their guidance and helpful feedback. We also wish to thank members of the Banca d'Italia present during our joint workshop, as well as members of the Machine Learning and Big Data Network of the Central Bank of Malta, for their comments and feedback.

 Correspondence: publicrelations@centralbankmalta.org.

[2] Senior Economist Statistician, Statistics Office, Central Bank of Malta.

[3] Economist Statistician, Statistics Office, Central Bank of Malta.

Abstract

This paper presents a Machine Learning approach adopted at the Statistics Department of the Central Bank of Malta to detect outliers in the Maltese Pension Schemes and Insurance datasets, which are collected by the Bank, at micro-level. The motive behind this study is to develop an outlier detection model which can detect outliers in both short and long time series. The model used to detect these anomalies in our data is the unsupervised Machine Learning Algorithm known as Isolation Forests. The model is used to capture large discrepancies between submissions for balance sheet data. The algorithm is implemented using KNIME Analytics Platform, with the use of the KNIME/Python integration feature. We describe our results using various tables and graphs, discuss the effectiveness and efficiency of the tool, and conclude by explaining some limitations and further improvements.

Contents

1. Introduction

The Statistics Department of the Central Bank of Malta continuously collects data from many different financial institutions. The aim of the statisticians within the Department is to ensure that the data collected is of high quality prior to being analyzed and disseminated. Entity-by-entity data collected from the reporting financial institutions may sometimes lack data quality. This may be due to various reasons, such as human errors or system errors, and this low-quality data creates problems to analysts assessing both the entity-by-entity and aggregated statistics.

Outliers are data points that differ significantly in value from other observations of the same reported item. In this case study, a reported item is a balance sheet variable, such as total assets. Taking this variable as an example, the total assets for each available reference date will be compared together for each Insurance Undertaking or Pension Scheme. We consider an outlier to be significantly different from the rest of the values depending on the threshold chosen in the Isolation Forest Algorithm.

It is important to detect and validate these outliers to ensure data quality and confirm the validity of substantial increases or decreases in values. Thus, one of the aims of a statistician is to detect these outliers, especially the large outliers that bring about drastic changes to the data when compared to the current or past data, at the data collection stage and to request the reporting institutions to provide information about these outliers.

Ideally, due to the large volumes of data being collected for the different reporting institutions, such data anomalies are identified in an automated manner. This reduces the burden on the data collectors and the latter would only be required to extract the collected data and to feed it into an automated process. In this way, the outliers would be flagged automatically and would therefore leave ample time for the data administrator to interpret. Furthermore, it is also vital to capture these anomalies in a rational manner and with the help of machine learning (ML) techniques, this procedure can be implemented effectively. Machine learning is a form of artificial intelligence (AI) that allows machines to learn, based on past and present experiences, by means of identifying patterns in the data fed to it. In this paper, since labeled data are not used, the authors have opted to make use of unsupervised techniques. 'Isolation forests' is one such technique, which will be used in this paper. This method immediately identifies anomalies instead of classifying normal instances. This technique is known to work efficiently for both small and large data sets. In the case presented in this paper, this was ideal since a vast range of data points will be dealt with.

The KNIME Analytics Platform was thought to be the best solution to implement this study. It has the advantage of handling big amounts of data, as well as being more user-friendly since it does not require much coding. Rather, it makes use of built-in nodes which only need to be configured once

by the user. Each node has a specific function which helps in creating a simple yet effective workflow process. Another major advantage of using this platform is that once the workflow is completed, the results are outputted with a click of a button each time new data is collected. In this project, such tool was expected to be able to capture instances where the anomalies are due to large increases or decreases compared with the rest of the values for a certain data point.

This paper will be divided as follows: Firstly, the theory behind the Isolation Forest (ML) method, and its advantages will be explained and an explanation on how it works in detecting the different outliers will be provided. Then, the paper will proceed to explain the data to be used for putting this technique into practice. The paper shall also explain the approach taken in KNIME to create a workflow process that first transforms the data as required, and then how the Isolation Forest algorithm was implemented on the transformed data to output the anomalies. In doing so, the paper will explain how Python was integrated in KNIME to be able to use this ML technique. The thinking behind the parameters of choice, the analysis, and the interpretation of the results obtained will also be explained. Finally, in the conclusion, this paper will point out limitations encountered throughout this project and the suggestion of improvements that can be made in the future to create a better and more efficient model.

2. Literature Review

Outlier detection is a critical task in data analysis, particularly in financial datasets where anomalies can indicate errors, fraud, or significant but rare events. Various machine learning methods have been developed to automate and improve the accuracy of outlier detection. This literature review explores the landscape of machine learning approaches to outlier detection, justifying the choice of Isolation Forests as a favourable method for detecting outliers in Maltese Pension Schemes and Insurance datasets.

The Local Outlier Factor (LOF) method is another prominent approach for outlier detection. LOF measures the local density deviation of a given data point with respect to its neighbours, identifying points that have a substantially lower density than their neighbours as outliers (Breunig et al., 2000). This method is particularly effective in detecting outliers in datasets with varying densities, making it suitable for complex datasets where outliers are not globally isolated but locally. LOF has been successfully applied in various domains, such as network intrusion detection and fraud detection, demonstrating its versatility (Aggarwal, 2013). However, LOF has several limitations. It requires careful tuning of the number of neighbours parameter (k), which significantly affects the performance and results. Additionally, LOF's computational complexity increases with the size of the dataset, making it less efficient for large-scale data. Furthermore, its reliance on local density measures means that it may struggle with high-dimensional

data, where the concept of "neighbourhood" becomes less meaningful due to the curse of dimensionality (Campos et al., 2016).

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a clustering method that identifies outliers as points that do not belong to any cluster (Ester et al., 1996; Campello et al., 2013). DBSCAN is advantageous for its ability to detect clusters of arbitrary shape and its robustness to noise. However, it also has significant drawbacks. The performance heavily depends on the choice of parameters (epsilon and minimum points), it assumes a constant density of points within clusters which may not hold true for financial datasets with varying distributions, and it struggles with very large datasets due to its quadratic time complexity in the worst case.

Isolation Forests were introduced by Liu et al. (2008) and have since gained traction as a robust and efficient method for outlier detection. Isolation Forests, an ensemble-based method specifically designed for outlier detection, address many limitations of the aforementioned methods. The core idea behind Isolation Forests is that outliers are few and different, and thus they can be isolated more easily than normal points. The algorithm works by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature. This process is repeated to create a forest of trees, with the path length from the root to a terminating node being shorter for anomalies, making them easier to detect (Liu et al., 2008; Hariri et al., 2019).

Several studies have demonstrated the effectiveness of Isolation Forests in various domains. In fraud detection, for instance, they have been employed to identify unusual patterns in transaction data, which is crucial for preventing financial losses (Bardet et al., 2019). In credit scoring, Isolation Forests have been used to detect outliers that may indicate potential defaulters, thereby aiding in risk assessment. Additionally, in transaction monitoring, they help in identifying anomalous activities that could signify fraudulent behaviour or operational errors (Saxena et al., 2020).

Isolation Forests have several advantages. They have a linear time complexity, making them highly scalable to large datasets, which is critical for financial data analysis. They require fewer parameters (number of trees and sub-sampling size), simplifying the model tuning process. Additionally, they do not assume any data distribution, making them versatile for various types of datasets. The anomaly score provided by Isolation Forests is intuitive, as it directly relates to the path length for isolation in the trees, enhancing interpretability. Their ability to handle large-scale data and identify subtle anomalies makes them particularly suitable for the analysis of Pension Schemes and Insurance datasets, where both short-term and long-term anomalies need to be detected.

3. Concept and Methodology

The unsupervised ML technique of Isolation Forests (Liu et al., 2008) is built with the use of multiple decision trees. A decision tree creates data models by means of following decision rules learnt from the features of the data. Moreover, the value of the target variable is predicted based on the formation of the tree. The trees are formed in such a way that random features are selected and split according to a random value contained in the feature selected. This can be seen graphically in Figure 1. An internal node (or non-leaf node) is defined as a node which contains data points that still need to be classified further. In other words, an internal node is a node performing a test on a particular attribute. An external node (or leaf node) is defined as a node which will not be split any further in the tree. In other words, an external node is an end node.

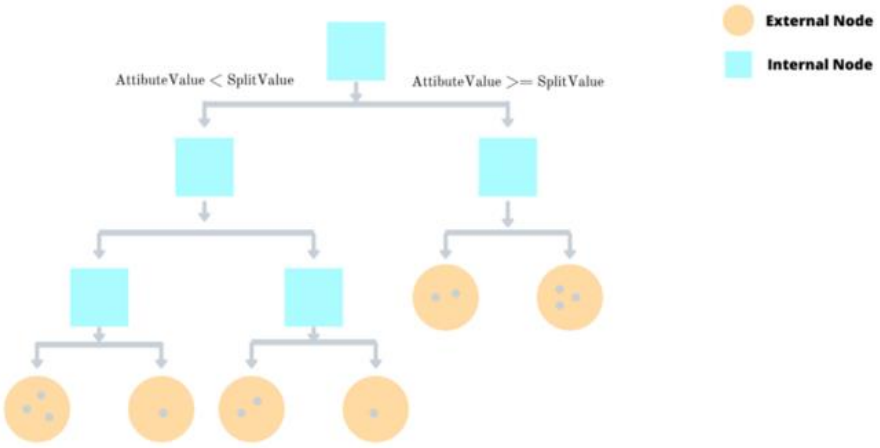

Figure 1: Decision Tree of an Isolation Forest

This splitting of data points into partitions is repeated until all instances are isolated into external nodes for each tree in the forest. In the case of outliers, it is likely that they are isolated early in the formation of the tree since anomalies are instances with distinguishable attribute values. As a result, outliers form shorter paths in the structure of the tree. A path is defined as a sequence of nodes that begin from the root node and travel from node to node along the tree. The path length $h(x)$ of a point x is measured by the number of nodes the point x travels to in the decision tree until travel is terminated by arriving to an external node. Moreover, isolated points are found close to the root of the tree, while inliers are more likely to be found at the deeper end of the tree.

As stated previously, an isolation forest is formed by building a collection of decision trees. When a substantial amount of random decision trees form

short paths for particular data points, it is likely that these points are outliers. From Figure 2, it can be seen that more partitions are needed in order to classify an inlier. Furthermore, we can consider the number of partitions to be equal to the length of the path from the root node to an end node which is required to isolate a data point. In Figure 2, we can see that the path length of point x_i is greater than that of x_0 since it requires more splits to isolate the data point.

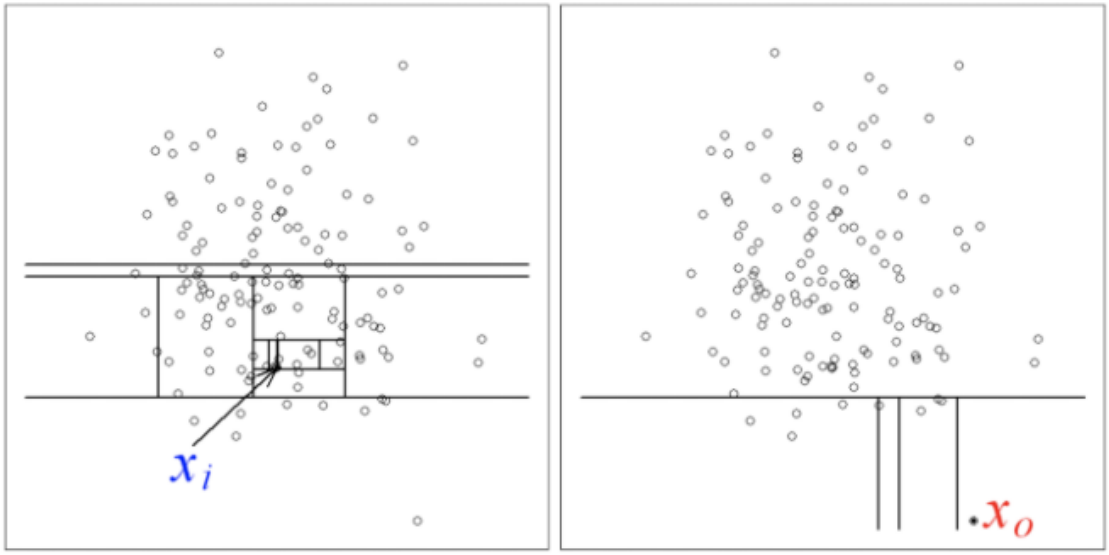

Figure 2: Partitioning of Data Points (Liu et al, 2008)

In theory, an outlier score $s(x,n)$ (Liu et al., 2008) is required to determine whether a data point is an anomaly or not. This score is calculated as follows:

$$s(x,n) = -2^{-\frac{\mathrm{E}(h(x))}{c(n)}},$$

where $h(x)$ is the path length of the data point under consideration x.

We are taking the average path length, denoted by $\mathrm{E}(h(x))$ over all the trees in the forest. $\mathrm{E}(h(x))$ is calculated by taking the path length $h(x)$ of every tree for point x, and dividing it by the number of trees in the forest.

Here, n denotes the number of external nodes in the decision trees which do not contain data point x. $c(n)$ is a special case of $\mathrm{E}(h(x))$, denoting the average path length of the external nodes. The respective score $s(x,n)$ may be interpreted as follows:

1) Close to -1: indicates that x is an outlier

For example, consider the extreme case when $\mathrm{E}(h(x)) = 0$. In this case the score becomes:

$$s(x,n) = -2^{-\frac{0}{c(n)}}$$

Therefore, $s(x,n) = -1$.

2) Much smaller than -0.5: indicates that x is an inlier

For example, consider the extreme case when $\mathrm{E}(h(x)) = \infty$. In this case the score becomes:

$$s(x,n) = -2^{-\frac{\infty}{c(n)}}$$

Therefore, $s(x,n) = 0$.

3) If all scores are close to -0.5: indicates that the entire sample does not seem to have clearly distinct anomalies.

The advantage of an isolation forest-based approach is that it identifies anomalies instead of classifying normal data points. Moreover, using this technique, it is easier to identify isolated points since their attribute values are different from normal instances, and there are also fewer instances of outliers compared to normal data points. Most other anomaly detection techniques are designed to optimise the profiling of normal instances rather than the profiling of outliers. Consequently, the results of such techniques may cause normal instances to be classified falsely as anomalies, or it may fail in detecting an actual outlier.

4. Data and Results

4.1 Data Description

For this study, two datasets will be considered; data submitted to the Department by the (a) pension funds/schemes and (b) insurance undertakings, respectively. Note that the variables which were not useful for this study were eliminated in the data transformation process within the KNIME workflow.

A brief overview of the variables contained in these datasets:

4.1.1 Data collected from the Pension Funds/Schemes

For this dataset, data spans from the period December 2020 up until December 2021 and is on a quarterly basis. Moreover, since the large institutions report such data on a quarterly basis while the smaller

institutions report on an annual basis, only the former institutions were taken into consideration. In the sampled data, this therefore translates to approximately 20 institutions.

It must be pointed out that the data under study was extracted from the balance sheets reported by the pension funds/schemes. Moreover, such balance sheet data are further classified on a country, sector, and currency basis. To avoid double counting, only the subcomponents were taken into consideration, therefore the totals were excluded from the exercise.

REFERENCE_DATE: Reporting period
ENTITYID: Scheme identifier
DENCOU/ CURRENCY: Currency
RESCOU/ COUNTRY: Country
SECCOUNSO/ SECTOR: Sector
TYPINS/ TYPE_OF_PHENOMENON: Balance sheet item (e.g. shares, loans)
CLOSING_POSITION: Stock of the item at the end of the reporting period

Table 1: Pension Fund Data Field Descriptions

The target variable for the anomaly detection process will be the 'Closing Position' variable. Note that in the case of datasets where flows (transactions, reclassifications and revaluations) are reported, the target variable can also be set to detect anomalies in the flows.

4.1.2 Data on the Insurance Undertakings

The insurance undertakings reporting return considered in this paper is the quarterly national specific template (CMQNT) and considers data starting from the period March 2016 up until December 2021. The data was collected by the Department every quarter. For this dataset, all the undertakings reported on a quarterly basis, therefore the whole population was taken into consideration. This amounted to approximately 85 reporting institutions in the sampled data.

It must be pointed out that the data under study was extracted from the balance sheet data reported by the undertakings. To avoid double counting, again the subcomponents only were taken into consideration.

REFERENCE_DATE: Reporting period
ENTITYID: Scheme identifier
TYPINS/ TYPE_OF_PHENOMENON: Balance sheet item (e.g. shares, loans)
SOLVII: The stock of the balance sheet item at the end of the reporting period (Solvency II Accounting Principle)
MGTACC: Stock of the item at the end of the reporting period (Management Accounts Accounting Principle)

Table 2: Insurance Data Field Descriptions

The target variable for the anomaly detection process was both the 'SOLVII' and the 'MGTACC' variables since the insurance sector balance sheet data are collected based on two accounting principles. Consequently, any data entries that did not include reported figures for both the 'SOLVII' and the 'MGTACC' variables were disregarded from the analysis. Furthermore, for data entries which included reported figures for the 'SOLVII' variable but not for the 'MGTACC' variable, and vice versa, the target variable was taken to be the variable under which data was reported. For data entries in which both the 'SOLVII' and the 'MGTACC' variables included reported figures, both variables were targeted for outlier detection.

4.2 Analyzing the Results

In this section, the results of our automated anomaly detection tool will be analyzed for both the insurance undertakings and the pension funds data. Firstly, the score threshold will be varied to find out how many outliers are being captured for these different thresholds. This will be done for both the pension funds and the insurance undertakings data. The optimal thresholds for both datasets will be then chosen and will illustrate the different scenarios that the model will capture in the outlier detection process so as to show the power and effectiveness of this tool.

PENSION FUNDS		
Threshold	Number of outliers detected by model	Contamination Percentage (%)
- 0.45 to - 0.55	2930	49
- 0.55 to - 0.65	501	8
- 0.65 to - 0.70	217	4

Table 3: Pension Funds Threshold Analysis Table

INSURANCE UNDERTAKINGS		
Threshold	Number of outliers detected by model	Contamination Percentage (%)
- 0.50 to - 0.60	16060	44
- 0.60 to - 0.70	4686	12
- 0.70 to - 0.80	1156	3
- 0.80 to - 0.85	150	0.4

Table 4: Insurance Undertakings Threshold Analysis Table

From the two tables above, it can be seen that as the score threshold is decreased, the number of anomalies captured also decreases. This is because the closer the score is to -1, the more likely that the data point is classified as an outlier. During this analysis, how the tool will work in practice was taken in consideration by choosing the optimal threshold, based on the right balance between practicality and the sensitivity of the tool. In practice, the user will not be able to go through thousands of data points during the screening process. Also, a high threshold resulted in many data points being mistakenly classified as outliers. Moreover, from the analysis, and taking into consideration the arguments above, the optimal thresholds were chosen as follows:

PENSION FUNDS		
Threshold	Number of outliers detected by model	Contamination Percentage (%)
- 0.65 to - 0.70	217	4

Table 5: Pension Funds Threshold Analysis Table

INSURANCE UNDERTAKINGS		
Threshold	Number of outliers detected by model	Contamination Percentage (%)
- 0.80 to - 0.85	150	0.4

Table 6: Insurance Threshold Analysis Table

In Tables 3, 4, 5 and 6, the contamination percentage column was computed by taking the ratio of the number of outliers flagged by the anomaly detection tool with the total number of data points under consideration, based on the threshold of choice. As explained above, it can be noticed that the contamination percentage decreases as the threshold decreases towards -1.

The following constitutes some examples of the anomalies detected using the above thresholds.

4.2.1 Data for the Pension Funds

Case 1

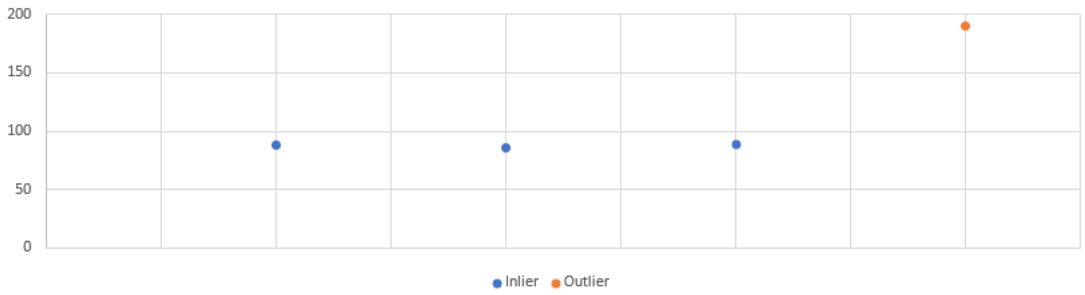

Figure 3: Scatter Plot for Case 1 - Outlier Detection in Pension Funds Data

The first case shows the detection of an outlier in a setting of small amounts. In this case, the outlier is evident. However, the next case will show that the tool is also capable of capturing less obvious outliers.

Case 2

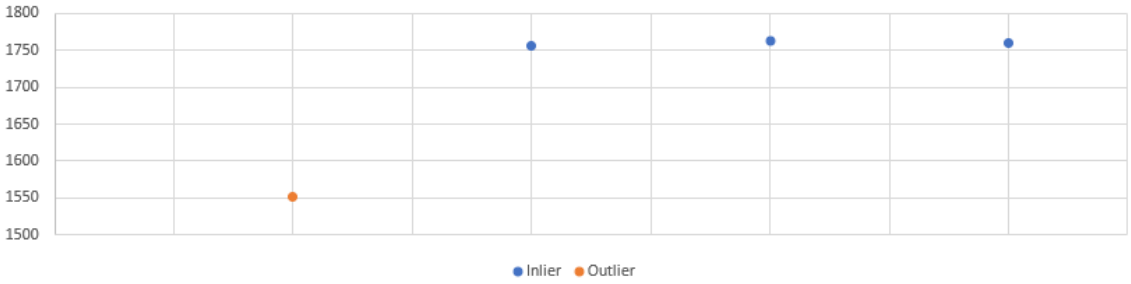

Figure 4: Scatter Plot for Case 2 - Outlier Detection in Pension Funds Data

This second case shows a setting where a less obvious outlier was successfully captured by the anomaly detection tool. In this case, the percentage change from the outlier period to the next period is only around 11%. Furthermore, this outlier was detected with only four data points, also showing the power of this tool.

4.2.2 Data for the Insurance Undertakings

Case 1

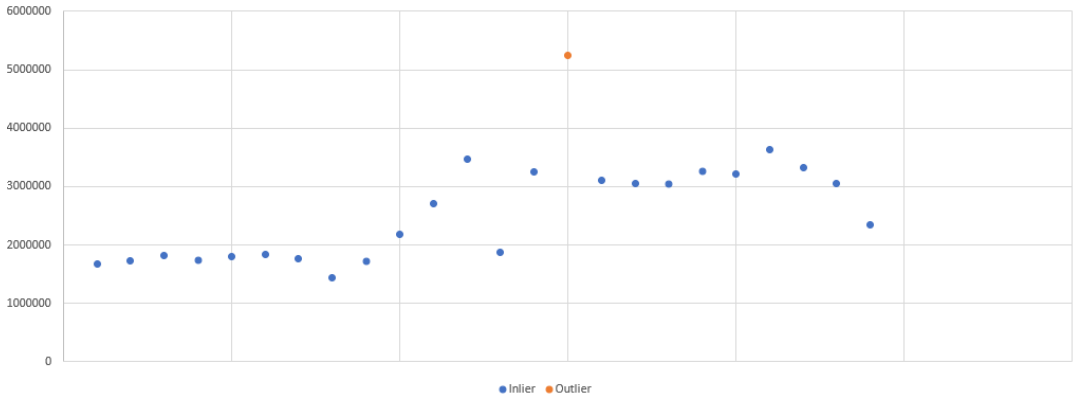

Figure 5: Scatter Plot for Case 1 - Outlier Detection in Insurance Data (for SOLVII Value)

The outlier in Figure 5 was confirmed as the correct closing balance by the reporting agent, despite being detected as an outlier.

Case 2

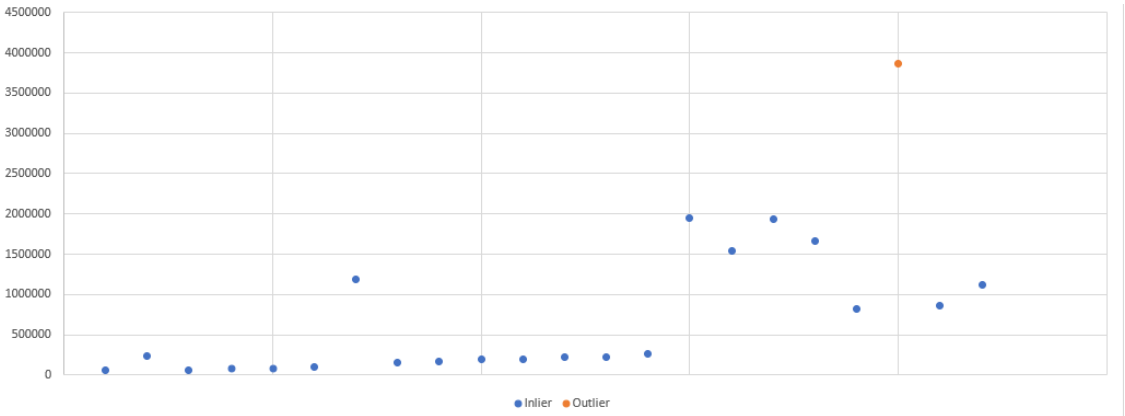

Figure 6: Scatter Plot for Case 2 - Outlier Detection in Insurance Data (for MGTACC Value)

The outlier in Figure 6 was confirmed by the respective reporting agent.

Case 3

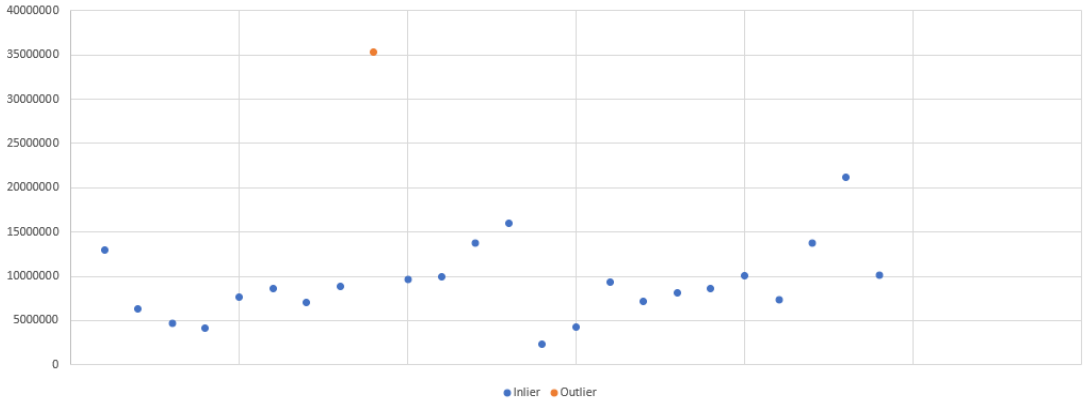

Figure 7: Scatter Plot for Case 3 - Outlier Detection in Insurance Data (for MGTACC Value)

The outlier in Figure 7 was also confirmed by the reporting agent, despite being captured by the tool as an outlier.

For the insurance sector data, the above cases show the detection of an outlier in a setting where the reporting institution was consistently reporting the target value for this data item but suddenly, for some reason, reported it as a slightly higher figure. Following this sudden increase, the consequent quarterly data were once again reported similarly as before. The anomaly detection tool detected such an inconsistency and flagged the higher value as an outlier.

Increasing the Score Threshold

This part of the paper will provide some examples of outliers detected when increasing the threshold to -0.6 for the pension funds data, and -0.75 for the insurance undertakings data. This will be done to show that the model captures the less obvious anomalies by changing the sensitivity in the model. As discussed above, choosing such higher thresholds was not suggested as it would not be practical, and it would be very time-consuming for the user to go through all the outputted outliers.

Case 1 – Pension Funds

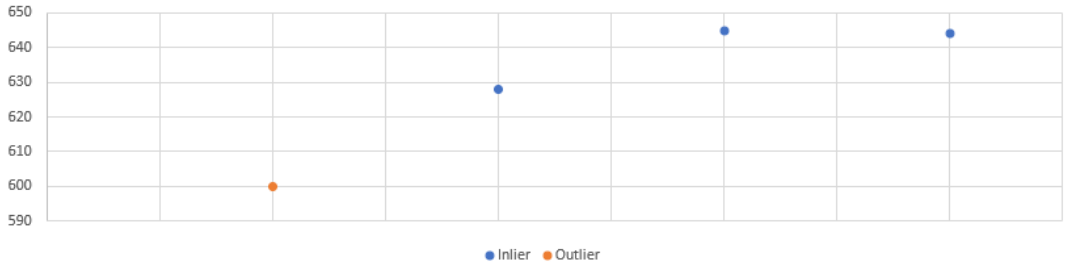

Figure 8: Outlier Detection for Pension Funds Data with Increased Threshold

Case 2 – Pension Funds

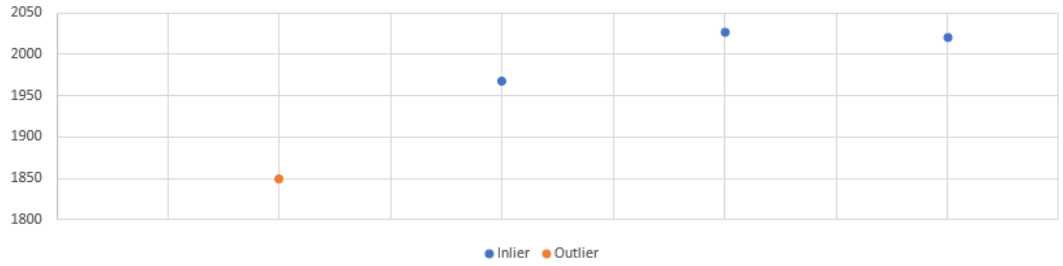

Figure 9: Outlier Detection for Pension Funds Data with Increased Threshold

Case 1 – Insurance Data

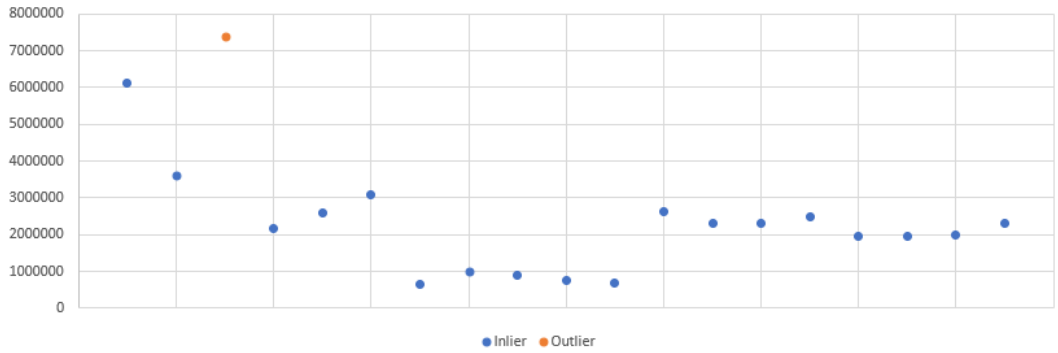

Figure 10: Outlier Detection for Insurance Data with Increased Threshold

The outlier detected in Figure 10 was confirmed by the reporting agent to be a correct figure, despite being detected by the tool as an outlier.

Case 2 – Insurance Data

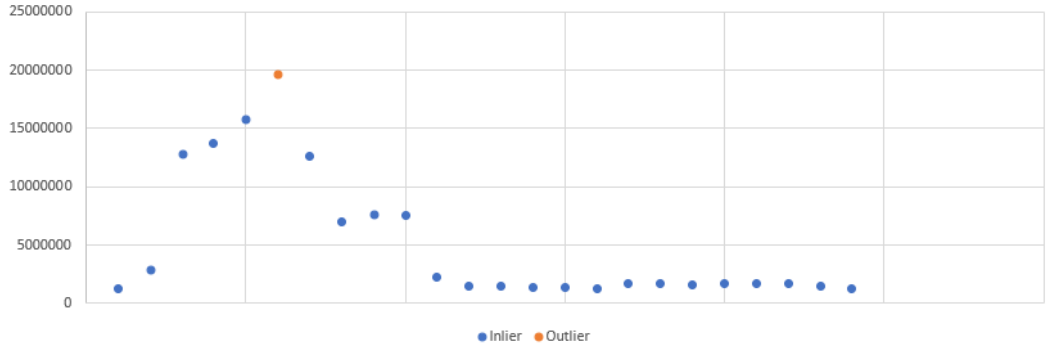

Figure 11: Outlier Detection for Insurance Data with Increased Threshold

Despite being detected by our tool, the outlier illustrated in Figure 11 was confirmed by the reporting agent to be correct. This means that the sudden increase detected by our tool was valid.

From the above figures, one can observe how the model captures data points and classifies them as anomalies even though they are not obvious outliers. This shows that the model also accommodates users who wish to increase the sensitivity in their analysis.

5. Conclusion

Despite the satisfactory results that were obtained, it is important to keep in mind that a machine learning model is never perfect and will always require improvements and fine tuning. In fact, in this analysis, a manual exercise was required to find the best parameters and score threshold. Furthermore, not every flagged data point was considered as an outlier in the results. Hence, human intervention is still required to perform a screening process of these anomalies. However, one of the biggest advantages of this tool is that the model continues to improve in detecting outliers as more data is fed into it since the model would have more information that can be used to help classify these data points.

One limitation in the results is that since pension funds data has not long started to be collected, only data for a relatively small number of entities was available, and for a small amount of reference periods. Hence, the model did not have much information to be able to learn how to classify results as optimally as it should. This will eventually be resolved once more data points are collected and fed into the model.

Another limitation of this outlier detection tool is that it does not take into consideration the time component. This means that when analysing our

data for outliers, the tool does not take into account any seasonality that may be present in our data set. To take such seasonality component into consideration, an anomaly detection method that takes into consideration the respective time component should be used.

6. Appendix

6.1 Application: KNIME Workflow

6.1.1 Reading, Cleaning and Transforming the Data

An Excel Reader node was applied since the raw data was in Excel format to read the data by providing it with the respective path from the local file system as seen in Figure 12.

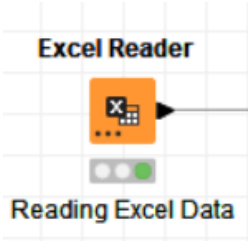

Figure 12: Excel Reader Node

The data cleaning and transformation steps are portrayed in Figure 13. After setting correct data types, and removing unnecessary rows and columns, a unique key is created. This is required to create groups of data in an attempt to detect outliers in a fair manner. For the pension funds data, the key was composed of the entity, type of instrument, country, sector, and currency, whilst for the insurance undertakings data the key was composed of the entity and the type of instrument.

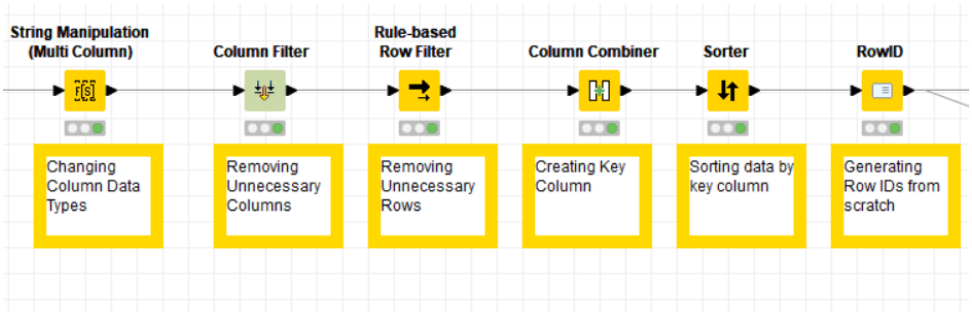

Figure 13: Data Cleaning and Transformation Nodes

6.1.2 Implementing the Isolation Forest Algorithm

The next step in the workflow involved making use of the Isolation Forest algorithm. Unfortunately, KNIME does not offer a node that executes such an algorithm directly on the platform. Consequently, Python was integrated in KNIME to make specific use of this anomaly detection algorithm using the generic Python Script node.

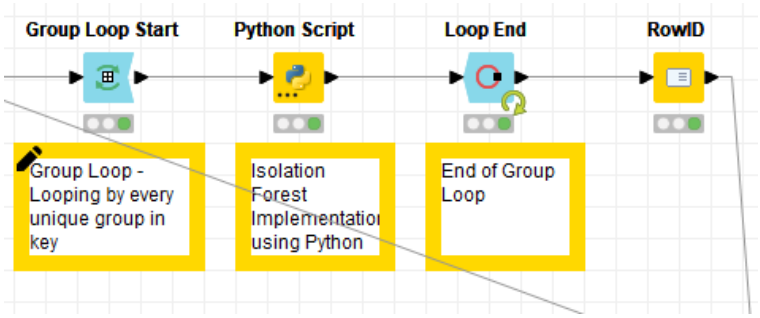

Figure 14: Group Loop and Isolation Forest Algorithm Nodes

Here, the Isolation Forest Algorithm was executed for every combination of pension scheme, currency, sector, country, and type of instrument. This combination is represented by the 'Key' variable, and the Isolation Forest algorithm is being looped on this 'Key' field with the use of the 'Group Loop Start' node and 'Loop End' node. This is illustrated in Figure 14.

The node 'Python Script' allows the user to insert and execute a Python script with the use of the KNIME-Python integration. In this script, the Isolation Forest package was imported into the Python environment. The variable 'CLOSING_POSITION' was set as the target variable to fit the Isolation Forest and executed the algorithm on this variable. Note that the parameters for the IsolationForest() method were taken to be the default parameters, as these were the most appropriate for the study. Since the aim of this tool was to be used in the day-to-day operations for all sectors in the Statistics Department, it would have been infeasible to perform parametrization for each sector.

Once the model was executed, the algorithm assigned a score to each data point according to every key combination. From this score one may then determine whether the data point is an outlier or not.

6.1.2 Outputting and Storing the Results

The resulting score values may vary between 0 and -1. The closer the score is to -1, the more likely it is an outlier. After analyzing the scores, it was concluded that the best threshold value would be -0.67. This means that

any data point whose score is less than -0.67, will be labelled as an outlier. Following this analysis, the 'Rule Engine' node was used to automatically classify the anomalies based on this chosen threshold. The configuration for this 'Rule Engine' Node is depicted in Figure 15.

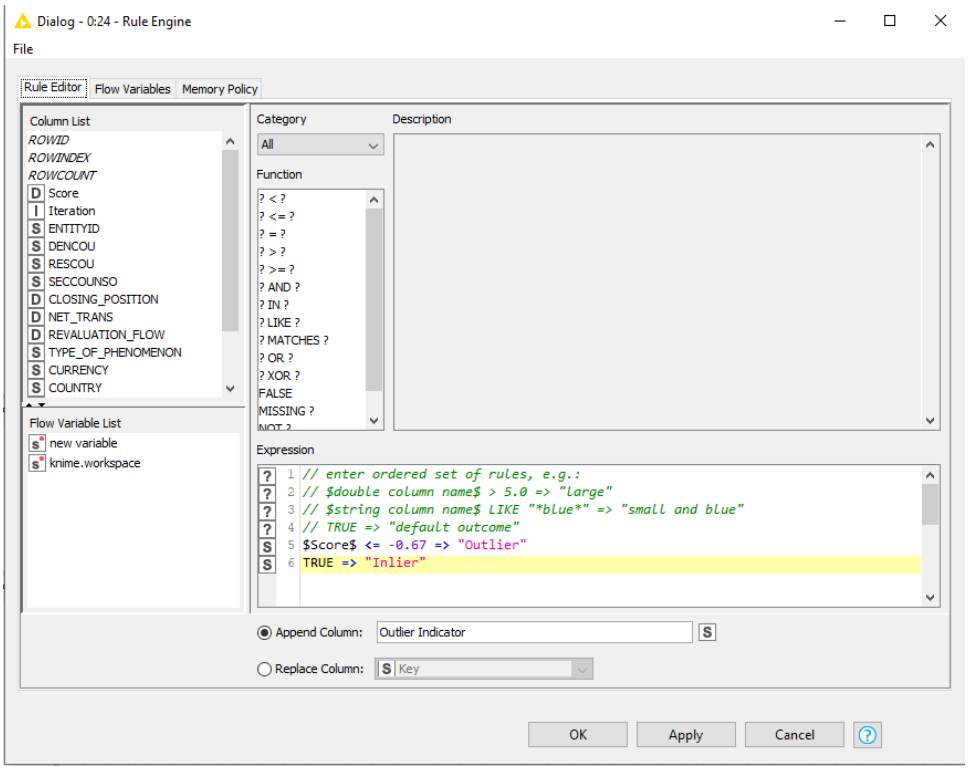

Figure 15: Rule Engine Node Configuration

Since the aim is to detect those data points which are anomalies, the 'Rule-based Row Filter' was used to output a table consisting of outliers only. Finally, the 'Excel Writer' node was used to output the results in Excel format.

Note that the workflow for the Insurance Undertakings data was built in the same way, with slight changes to cater for the differences in the data structure of the target columns.

7. References

Aggarwal, C. C., 2013. *Outlier Analysis*. Springer.

Bardet, J. B., Lang, G., Mameri, A., & Ould Said, E., 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*. Springer, Cham, pp. 253-267.

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. ACM, pp. 93-104.

Campello, R. J. G. B., Moulavi, D., & Sander, J., 2013. Density-based clustering based on hierarchical density estimates. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, Berlin, Heidelberg, pp. 160-172.

Campos, G. O., Zimek, A., Sander, J., Campello, R. J. G. B., Micenková, B., Schubert, E., Assent, I., & Houle, M. E., 2016. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, 30(4), pp. 891-927.

Ester, M., Kriegel, H. P., Sander, J., & Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *KDD*. pp. 226-231.

Friday talks: The dark horse of Isolation Forest. (2022). Retrieved 28 June 2022, from https://thingsolver.com/friday-talks-the-dark-horse-of-isolation-forest/

Hariri, S., Kind, M., & Brunner, R. J., 2019. Extended Isolation Forest. *arXiv preprint arXiv:1811.02141*.

Isolation Forest algorithm for anomaly detection. (2022). Retrieved 28 June 2022, from https://medium.com/@arpitbhayani/isolation-forest-algorithm-for-anomaly-detection-f88af2d5518d

KNIME AG, S. (2022). KNIME Python Integration Installation Guide. Retrieved 28 June 2022, from https://docs.knime.com/2019-12/python_installation_guide/index.html#setup_python_integration

Liu, F. T., Ting, K. M., & Zhou, Z. H., 2008. Isolation forest. In: *2008 Eighth IEEE International Conference on Data Mining*. IEEE.

Outlier Detection with Isolation Forest. (2022). Retrieved 28 June 2022, from https://towardsdatascience.com/outlier-detection-with-isolation-forest-3d190448d45e

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*. Springer, Singapore, pp. 231-242.

Setting up the KNIME Python extension. Revisited for Python 3.0 and 2.0 | KNIME. (2022). Retrieved 28 June 2022, from https://www.knime.com/blog/setting-up-the-knime-python-extension-revisited-for-python-30-and-20

sklearn.ensemble.IsolationForest. (2022). Retrieved 28 June 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html