Anomaly Detection applied to Money Laundering Detection using Ensemble Learning

Daniel Otero Gómez[1]

Santiago Cartagena Agudelo[2]

Andrés Ospina Patiño[3]

Advisor:
Edgar Lopez Rojas [4]

Research proposal
Mathematical Engineering
Department of Mathematical Sciences
School of Sciences
Universidad EAFIT

December 2021

[1] Mathematical Engineering Student at Universidad EAFIT (CvLAC: Daniel Otero Gómez)
[2] Mathematical Engineering Student at Universidad EAFIT (CvLAC: Santiago Cartagena Agudelo)
[3] Mathematical Engineering Student at Universidad EAFIT (CvLAC: Andrés Ospina Patiño)
[4] CEO founder at FinCrime Dynamics edgarlopez@fincrimedynamics.com

1 Problem Statement

Financial crime and, specifically, the illegal business of money laundering are increasing dramatically with the expansion of modern technology and global communication, resulting in the loss of billions of dollars worldwide each year. Money laundering, known as the process that transforms the proceeds of crime into clean legitime assets, is a common phenomenon that occurs around the world. Irregular obtained money is generally cleaned up thanks to transfers involving banks or companies, see Walker (1999). Hence, one of the main problems remains to find an efficient way to identify suspicious actors and transactions, in each operation attention should be focused on the type, amount, motive, frequency, and consistency with the previous activity and the geographic area. This identification must be the result of a process that cannot be based solely on individual judgments but must, at least in part, be automated. Although prevention technologies are the best way to reduce fraud, fraudsters are adaptive and, given time, will usually find ways to overcome such measures, see Perols (2011). Then, what we propose is to enrich this set of information by building an anomaly detection model in operations related to money transfer in order to benefit from the power of artificial intelligence. Now, anti-money laundering is a complex problem but We believe Artificial Intelligence can play a powerful role in this area.

It is important to note that this research will be guided by the data science organization of EAFIT University named Google Developer Student Club EAFIT (GDSCE) in conjunction with FinCrime Dynamics, which is an Agent-based simulation and analytics company that is dedicated to finding solutions to problems related to financial fraud and money laundering. Collaboration by the mentioned company will be based on advisory and access to some of the tools developed by the company.

2 Objectives

2.1 General objective

To implement an anomaly detection framework in favor of verifying how good it is with respect to Precision and Recall.

2.2 Specific objectives

- To carry-out an exploratory data analysis that captures data relations through metrics and visualizations.
- To perform feature engineering and feature selection with the information found within the anomaly detection.
- To implement and evaluate anomaly detection algorithms that detect suspicious transactions.

3 Methodology

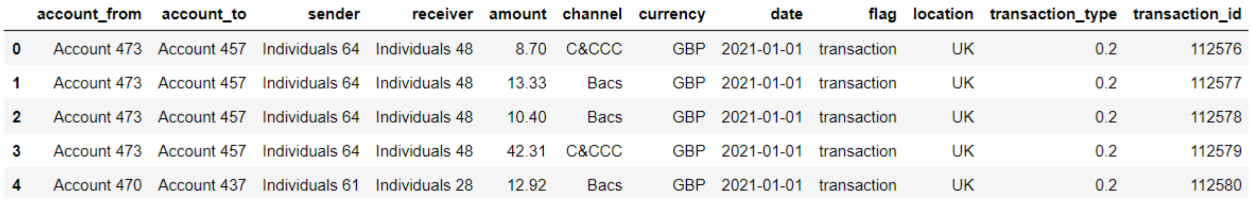
The methodology followed throughout this work was the Cross-Industry Standard Process for Data Science Mining (CRISP-DM). Regardless of whether the data science work to be done is large or small, it is always useful to apply techniques and strategies that help with its planning, development, and maintenance. This scheme divided the process of problem-solving into six different stages: Business Understanding, Data Understanding, Data Preparation, Modelling Phase, Evaluation, and Implementation. The objective of offering such a sequence was to provide a basis with which tasks may be decomposed from general to specific, without imposing a way to perform them.

4 State of the Art

Currently, outlier detection refers to the problem of the identification and, where appropriate, the removal of anomalous observations from data. There is no official definition of what constitutes an outlier; they can be broadly seen as observations that deviate enough from the majority of observations in a dataset to be considered the product of a different generative process. Hence, given a dataset, the percentage of outlier observations is usually small, typically lower than 5%, see Domingues *et al.* (2018). A significant part of the literature focuses on the undesired properties of outliers; they can nevertheless reveal valuable information about previously unknown characteristics of the systems and entities that generated them, see Boukerche *et al.* (2020). Existing outlier detection methods have been proven to be efficient for a diverse pool of applications, including credit card fraud detection, money transactions, network intrusions, and many others that require the processing of high-dimensional data or huge amounts of data streams, Domingues *et al.* (2018). The removal of outlier observations from data is also to the benefit of machine learning and statistical modeling, as an outlier-free dataset enables such algorithms to capture the emerging trends more accurately. According to Meng *et al.* (2019), there are two main machine learning approaches for the problem of outlier detection. The first approach, unsupervised outlier detection, assumes little prior knowledge of the data. Under this approach, unlabeled data are split into clusters and any observations separated from the main clusters are flagged as potential outliers. The second approach, supervised outlier detection, tries to explicitly model and learn what constitutes an outlier and what separates an outlier from normal observations. The last approach is considered related to semi-supervised classification. While plenty of unsupervised outlier methods are present in the existing literature, they usually put too much weight on a single a measure of ‘outlierness’ such as density or distance, ignoring other measures. Additionally, many feature selection strategies have been used when working with highly skewed data as the data obtained for the development of this work. The suggested methodology by Zimek & Filzmoser (2018) integrates existing outlier-detection techniques using a voting ensemble, but at the same time employs an unsupervised feature selection process in order to overcome the individual shortcomings and finally produce a single confidence score. Alelyani *et al.* (2018) showed that selecting subsets of features, according to some similarity or correlation criteria, can boost unsupervised learning algorithms analogously to how supervised learning algorithms are improved. Several unsupervised learning feature selection methods have been proposed for different kinds of data. To improve the detection accuracy and stability, researchers have recently devoted their efforts to the application of ensemble methods to outlier detection problems, and several new outlier ensemble algorithms have been proposed. Ensemble learning uses combinations of various base estimators to achieve more reliable and superior results than those attainable with an individual estimator, Zhao & Hryniewicki (2019).

5 The Dataset

As GDSC Eafit partnered with FinCrime Dynamics, a company who dedicates solely to developing solutions within the Financial Crime domain, we had access to an agent-based simulation tool that allowed us to generate as much data as we needed.



	account_from	account_to	sender	receiver	amount	channel	currency	date	flag	location	transaction_type	transaction_id
0	Account 473	Account 457	Individuals 64	Individuals 48	8.70	C&CCC	GBP	2021-01-01	transaction	UK	0.2	112576
1	Account 473	Account 457	Individuals 64	Individuals 48	13.33	Bacs	GBP	2021-01-01	transaction	UK	0.2	112577
2	Account 473	Account 457	Individuals 64	Individuals 48	10.40	Bacs	GBP	2021-01-01	transaction	UK	0.2	112578
3	Account 473	Account 457	Individuals 64	Individuals 48	42.31	C&CCC	GBP	2021-01-01	transaction	UK	0.2	112579
4	Account 470	Account 437	Individuals 61	Individuals 28	12.92	Bacs	GBP	2021-01-01	transaction	UK	0.2	112580

Figure 1: Sample of one of the generated datasets (property of FinCrime Dynamics).

Now, as it can be observed in Figure 1, the data set is composed mostly of categorical columns which suppose a big challenge since clustering and classification algorithms are built to work solely with numerical data (reason for which many algorithms work very poorly with one-hot encoded variables). The flag and transaction type are columns that correspond to information that is not generally known beforehand. The first one is a binary column that indicates whether the transaction is fraud or not (target variable), and the latter corresponds to a set of numerical values that map the motive of the transaction (i.e. domestic purchase). Since the location and currency variables contain a single value (the UK and GBP respectively) the columns are dropped. Furthermore, the channel variable corresponds to the medium in which the transaction was made (i.e. cash or a platform or company, Paym for instance).

We wanted to make the modeling phase as flexible as possible, so we decided to focus on enlarging the number of numerical columns during the feature engineering phase.

6 Exploratory Data Analysis

As it was stated previously, PaySim simulates data that is very similar to real-life behavior, however, several different data sets can be generated with it. The number of businesses, individuals, and agents may be varied to alter certain dataset characteristics such as the fraud percentage. And it is for this same reason that it is necessary to do an exploratory analysis, because as it is a simulation, the data may be biased by patterns

It was observed that the two types of transactions (individual and business) follow a heavily right-skewed distribution since most transactions are near to the lower whisker, that is to say, that most of the transactions are low amounts. There are a few numbers outliers when the transaction numbers are very high and if they are compared, it is possible to appreciate that there is much more number of transactions in the business type, which makes sense.

On the other hand, most transactions are low value and regardless of the channel, they follow a right-skewed distribution, which means that there are going to be fewer transactions the larger the amount of money transferred. In addition, it was identified that the number of transactions made in cash is much lower than the others, which follow a more uniform distribution.

It is noted that most transactions are between the same type of agents (individual to individual or business to business), while the number of transactions between different types is considerably less.

In general, was found a pattern in the analyzed data sets, specifically with the fraudulent transactions. Although the amount of fraudulent transactions is varied, it is possible to identify that certain amounts are repetitive, especially that of 20.000, moreover, it was identified that transactions of GBP2000, GBP20000, GBP120000 are always fraud transactions; Regarding the channel, it is possible to identify that the majority of transactions made in cash are fraudulent, in four of the five data sets about the 95% of the transactions made in cash are a fraud; And finally, curiously, the vast majority of fraudulent transactions occur on Fridays (4), about 80% of the fraudulent transactions are made this day.

Furthermore, we decided to implement more complex ensembles to achieve higher performance. Firstly, we constructed a bagging approach in which we trained a set of models and make predictions with them. The final prediction is generated according to the maximum vote criteria. This framework idea was discarded very quickly since the model performed poorly. We attributed the bad performance to the heterogeneity of the individual models. The number of transactions that were labeled as fraudulent was very low, which implied that the different models did not agree on which observations were anomalies or not.

7 Experimentation Methodology

Firstly, the work and experimentation were centered around feature engineering. As it was stated in the previous section, working with several categorical variables represented a big challenge for classical ML models. However, we attempted to build a method that allowed us to capture the information contained in the non-numerical values and represent it numerically. The chosen approach was built within the idea that it is possible to group transactions according to certain categorical criteria, for instance, in which month was made, and calculate the desired statistic that characterizes that group and assign it to every transaction that belongs to it. Since our goal was to work around the transacted amount, which intrinsically has a heavily right-skewed distribution, we decided to use robust statistics to avoid any outlier bias to affect the operations. Consequently, we built a function that groups the transactions according to a given set of categorical variables and calculates the Median and the Median Absolute Deviation (MAD) of every group. These statistics are not only used to map the values of each transaction but also to create unique values for each transaction. This was made by subtracting the median from the transacted amount and then dividing it by the MAD. This is a robust attempt to calculate the deviation from the median transaction and standardize it according to a robust deviation metric. Moreover, the function will output two variables, since the grouping will be made according to the type of sender and type of receiver of the transaction as well (which could be a person or a business). This will further characterize the behavior and the commonness of the transaction not only by the specified group but also by the type of individual that is making and receiving the operation. Since the data set counts with five relevant categorical variables (account from, account to, sender, receiver, and channel) the possible combinations are wide. We built up to 52 new variables with this method. However, more variables do not necessarily imply better performance. Many of these variables may not be relevant and feed noise to the models, thus, a featured technique is necessary to rank variables by relevance.

Regarding the feature selection, we found a method referenced as SPEC which is a unified framework, based on spectral graph theory, that enables the joint study of both supervised and unsupervised feature selection. The main idea of SPEC is to represent data points as vertices of a graph and assign weights to edges of the graph corresponding to the distance or measure of similarity between points—a commonly used similarity measure being the RBF kernel function. The SPEC framework selects features, by studying the degrees of similarities among samples. Under this framework, features consistent with the graph structure are assigned similar values to data points that are close to each other in the graph. Such features would be of increased relevance since they behave similarly in each similar group of samples, see Zhao & Liu (2007). After implementing other feature selection methods, and after some comparisons between them it was noticed that the SPEC method was the best since it was much more related to the nature of the problem while the others were used as a form of contrast. The SPEC results were not even shown for their power over the other feature selection methods that were implemented.

After this, efforts were focused on the modeling part, for which different tests were carried out with different kinds of models. Several modeling frameworks were used to achieve a good performance. Before explaining these frameworks it is important to clarify the evaluation procedure that was applied to the experimentation. This study focused on the utilization of unsupervised models, an approach that allows us to evaluate the training and testing error since these methods often produce their predictions by assigning an anomaly score to each observation and affirming that an event is anomalous if its anomaly score is above a certain threshold (which is often related to the data contamination percentage). The aforementioned occurs for the training and testing cases. Good performance will be reflected by achieving consistently good results across the two phases.

Now, regarding the implemented frameworks. Firstly, a single model framework was applied to make tests around feature engineering and feature selection and set a performance baseline for each of the combinations. The second considered framework was a bagging procedure, in which several models produce predictions over the data set and the majority vote scheme is used to decide whether a transaction is an anomaly or not. This framework will not be shown in the results since it was concluded very quickly that the framework is not useful. This occurs due to the heterogeneity of the predictions of the models, this causes that a very little amount of transactions are labeled as anomalous. Furthermore, these few anomalous transactions are mostly erroneous.

Alternatively, other ensemble frameworks were designed to improve the performance. The main idea relies on the heterogeneity of the predictions of different models and how to leverage this property. In the literature is found that performing stacking and representation learning are powerful approaches to the problem. Both methods share the idea of defining a set of base-learners that will make predictions about the data, in this case, assign anomaly scores. The stacking framework states that the scores generated are appended to the original data set as if they were new characteristics of the data. The representation learning framework differs from this in the way that the new scores are now used as a new data set. Finally, for both frameworks, the new data set is fed to a meta-learner that will be in charge of making the final prediction.

Additionally, in Zhao & Hryniewicki (2018) a filtering procedure of the scores assigned by each model is described. The method is constructed under the idea of preserving the most accurate, yet, less correlated scores. The purpose of the correlation idea behind this relates to the heterogeneity of the models mentioned before. A representation learning/stacking procedure must look to generate the new feature vector composed of unique values for each characteristic. If two models output very similar results, then the matrix contains redundant information and one of the variables must be removed. The same methodology is applied to both frameworks described previously and in the results section, it is evaluated if this has a significant improvement.

Finally, we decided to implement a repetitive filtering framework for both stacking and representation learning to see if it is possible to generate more representative new characteristics by performing deeper characteristic generation. This will be done by storing together the generated variables after every iteration, performing the filtering process over this new data set, and feeding the base learners with the data set with the appended new characteristics.

The models are evaluated by using a 5-fold cross validation. In the Results section the Area Under the Curve (AUC) of the Precision-Recall Curve, the Recall, the Precision, and the F1 score are reported. The main metrics with which we will be evaluating the performance of the model are the AUC and the Recall. The AUC reflects how well the precision and recall balance across different thresholds, giving information on how well the model performs across this two metrics. We considered recall as a single metric as well because we wanted to produce a model that has a good fraud detection rate, not only balance the precision of it.

The experiments described in this section were conducted using Jupyter Notebook and using PyOD (Python for Outlier Detection) and Scikit Learn as base libraries for modeling, framework construction, and evaluation automatization.

It is also important to highlight the emphasis that was made on the Isolation Forest. This model was used as a meta-learner in every step of the experimentation due to its robustness and great performance across different data sets (Liu *et al.*, 2008) Zimek *et al.* (2014). Additionally, after a thorough set of experiments, it was concluded that it is not necessarily good to use a wide number of different base learners to produce a better performance. It is sufficient with including a small number of different models that use different ways of estimating anomaly scores and include a few versions of each of these models using different parameters. This idea is used in DCSO, see Zhao & Hryniewicki (2019). After several tests, we decided to consider three versions of Principal Components Analysis (PCA), Cluster-Based Local Outlier Factor (CBLOF), and K-Nearest Neighbors (KNN) tuned with different hyperparameters as base learners. We assigned 3, 5, and 7 components for PCA; 3, 6, and 9 clusters for CBLOF; and 2,4, and 6 for KNN. We enabled sample bootstrapping and used a 200 tree ensemble for the Isolation Forest.

8 Results

Model	TR AUC	TS AUC	TR Recall	TS Recall	TR Precision	TS Precision	TR F1	TS F1
NF-Cont=0.01	0.12	0.18	0.67	0.51	0.2	0.16	0.31	0.24
F-Cont=0.01	0.12	0.16	0.65	0.51	0.2	0.16	0.3	0.24
F-Cont=0.005	0.16	0.12	0.51	0.65	0.16	0.2	0.24	0.3

Table 1: Comparison of different methodologies of single model framework.

As was pointed in the previous section, the experimentation process started with the comparison of different feature engineering and feature selection methodologies. In the case of feature selection, the experimentation was very quick, since techniques such as the variance threshold, the K-Best features, and tree feature importance analysis were immensely outperformed by the SPEC procedure. Moreover, the built feature engineering function allowed us to generate a large number of new variables to the data set. However, through testing of different models, it was noted that the models' performance was reduced when a larger set of variables was fed to them. Consequently, we tested different criteria to generate new variables and concluded that building general features had the best impact within the model performance, hence, the experimentation was conducted using combinations of less than three categorical variables; moreover, it was noted that the spectral feature selection improves on average 71.5% the mean AUC since it considers only those variables that provide relevant information and discards those that can become ambiguous information for the model, in addition to leading to improvements with increasing layers.

Now, regarding the model testing, we set a baseline by testing a single model approach using the Isolation Forest as a model. The results of three different basic configurations of the single approach framework are shown in Table 1. Only the learning of the model with the initial features is evaluated, with which the following was obtained; in the first place, the model was proved using all the features achieved with the feature engineering and the based learners with which bad results are obtained since many authentic transactions are considered as a fraud when they were not, leading to a low average AUC, so the next experiment was to make a spectral feature selection (1), and despite obtaining very similar results, it is significant because it was obtained with fewer features. Then, the contamination level was reduced (1), which led to a better precision but a worse recall, and a mean AUC like the previous ones.

Considering the low results, we proceeded to test with different types of frameworks, varying only the level of contamination, and the results improved considerably, especially with filtering (evaluating only the most relevant features).

Model	TR AUC	TS AUC	TR Recall	TS Recall	TR Precision	TS Precision	TR F1	TS F1
Data set	0.12	0.16	0.65	0.51	0.2	0.16	0.3	0.24
Stacking	0.17	0.23	0.67	0.59	0.2	0.18	0.31	0.28
Filtered Stacking	0.16	0.21	0.67	0.52	0.2	0.16	0.33	0.24
Representation Learning	0.34	0.34	0.67	0.68	0.2	0.21	0.31	0.32
Filtered Representation Learning	0.42	0.39	0.74	0.75	0.22	0.23	0.34	0.35

Table 2: No extra layers, Contamination=0.01

Model	TR AUC	TS AUC	TR Recall	TS Recall	TR Precision	TS Precision	TR F1	TS F1
Data set	0.16	0.12	0.51	0.65	0.16	0.2	0.24	0.3
Stacking	0.23	0.17	0.59	0.67	0.18	0.2	0.28	0.31
Filtered Stacking	0.21	0.17	0.52	0.67	0.16	0.2	0.24	0.31
Representation Learning	0.34	0.34	0.68	0.67	0.21	0.2	0.32	0.31
Filtered Representation Learning	0.38	0.42	0.75	0.74	0.23	0.22	0.35	0.34

Table 3: No extra layers, Contamination=0.005

Model	TR AUC	TS AUC	TR Recall	TS Recall	TR Precision	TS Precision	TR F1	TS F1
Data set	0.12	0.16	0.66	0.51	0.2	0.16	0.31	0.24
Stacking	0.18	0.23	0.67	0.65	0.2	0.2	0.31	0.3
Filtered Stacking	0.22	0.27	0.7	0.67	0.21	0.2	0.33	0.31
Representation Learning	0.33	0.36	0.67	0.67	0.2	0.2	0.31	0.31
Filtered Representation Learning	0.46	0.47	0.83	0.82	0.25	0.25	0.39	0.38

Table 4: Contamination = 0.01 layers = 2

Model	TR AUC	TS AUC	TR Recall	TS Recall	TR Precision	TS Precision	TR F1	TS F1
Data set	0.12	0.17	0.19	0.37	0.11	0.23	0.14	0.28
Stacking	0.18	0.24	0.55	0.5	0.33	0.31	0.42	0.38
Filtered Stacking	0.21	0.26	0.58	0.48	0.36	0.29	0.21	0.26
Representation Learning	0.35	0.34	0.67	0.60	0.41	0.38	0.51	0.46
Filtered Representation Learning	0.47	0.42	0.73	0.79	0.45	0.49	0.55	0.60

Table 5: Contamination = 0.005 layers = 2

In general, it can be seen that there were improvements when some other features were added to the data set, since in all the cases the stacking model gives better results, however, the best of the models is when are considered only those features generated by the based learner and filtered, which means that the models are managing to capture the most relevant information from the data set. Moreover, it is noted that decreasing the level of contamination only significantly affects the level of precision, although it is well known that this hyperparameter is of vital importance. In addition, the number of layers consider was vital for achieving better results, especially in the mean AUC and the Recall metrics, since the best model is achieved with a 0.01 level of contamination and two layers, considering only those features generated by the based learners. Finally, through experiments related to the number of layers, it was found that the model does not need a lot of layers to achieve good results, since after two layers the improvement is no longer significant.

In general, the best results are seen in filtering; The decrease in the level of contamination, taking into account that we are working with a data set in which approximately 0.3% of the data is classified as fraud, generates better results as far as precision is concerned, however, the recall decreases, also it is observed that the mean AUC still being better with a 0.01 contamination. Finally, it is observed that the layers improve the model because when increasing them, better results are obtained in all the metrics.

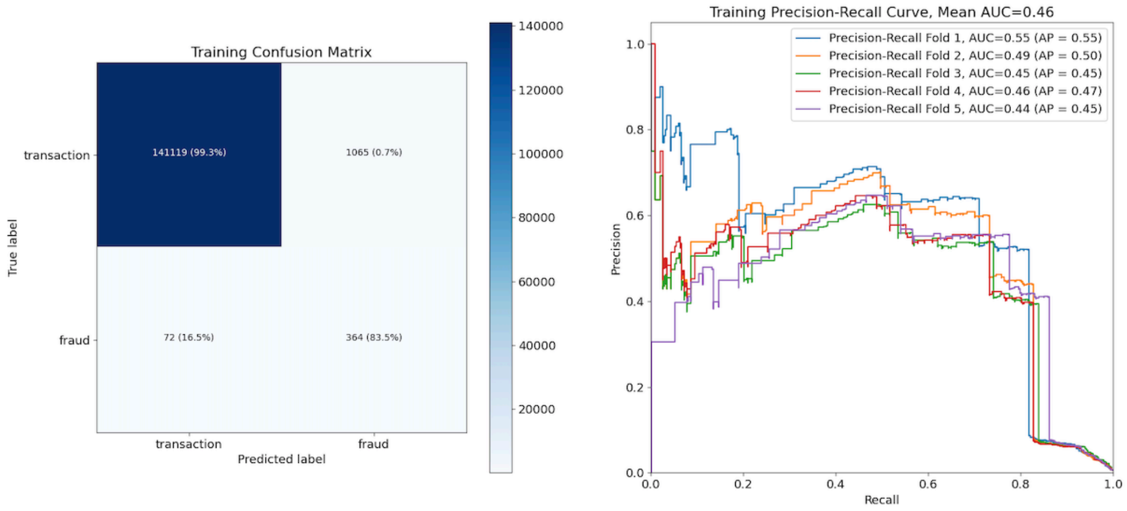


Figure 2: Training results. Contamination = 0.01, layers = 2

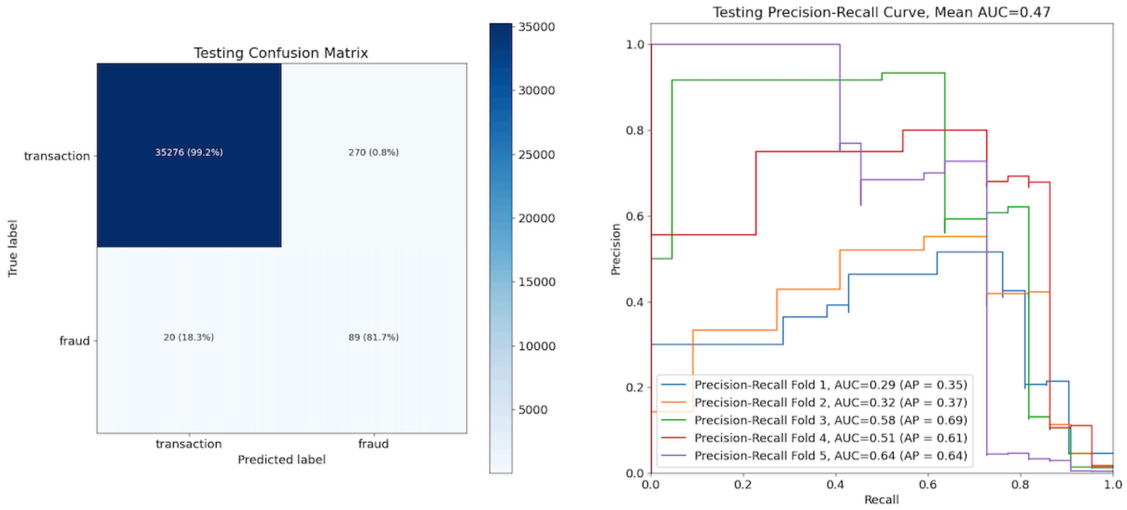


Figure 3: Testing results. Contamination = 0.01, layers = 2

From the previous graphs, it can be observed that in general, the training data present better results, which is to be expected, since in this it is possible to capture 4 out of every 5 fraudulent transactions approximately. Moreover, it is possible to recognize the good performance in the precision-recall graph since in many cases a precision of 0.8 or greater for high recall values. Despite observing some variance in the testing, it is reduced a bit when compared to the other models, however, there is still overfitting.

9 Schedule

Table 6: Schedule

Activity	*Weeks*															
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*
Literature Review	■	■	■	■												
Data Understanding and Preparation					■	■	■	■								
Feature Engineering and Feature Selection									■	■	■	■				
Classification and Evaluation													■	■	■	■

Literature Review: In this phase it consisted of conducting an in-depth search of the elements of the literature and the web that could help us when starting the execution of the project as such.

Data Understanding and Preparation: Drives the focus to identify, collect, prepare and analyze the dataset that will be used by describing, exploring, analyzing and verifying data quality.

Feature Engineering and Feature Selection: At this point the process of reducing the number of input variables is carried out when developing a predictive model, as well as the process of using domain knowledge to select and transform the most relevant variables from raw data when creating a predictive model using machine learning or statistical modeling.

Classification and Evaluation: In this part the task is to evaluate the results obtained from each model and compare them, check if all the models succeeded based on your criteria and once you choose the model or models to work with is time to determine the next steps.

The work was strict due to we had to keep constantly in touch with FinCrime Dynamics. This was in order to offer the best possible results given their needs and also to get feedback on what we were working. The actual execution of the plan was similar to the one presented in the pre-project, with the difference that we decided to use one more week for business understanding. The latter as a consequence of becoming more familiar with the company, their work and the tools they provided us with for the development of this work.

10 Ethical Implication

Financial crimes, like money laundering cost several billions of dollars per year and affect the lives of millions of people, because most of the time these laundered funds are used to foster further illegal activities such as the financing of terrorist activity, trafficking of illegal drugs, support of prostitution rings, or smuggling of weapons, see Sudjianto *et al.* (2010). In this way, this article wants a better understanding of how this type of fraud can be identified through machine learning.

11 Legal Aspects and Commercialization

Due to legal policies and politics that protect customers' privacy financial records, most of the time, this data is not accessible. That is why a confidentiality agreement (NDA) was signed that guarantees that confidential information is not disclosed to third parties. AI can effectively carry out tasks traditionally done by humans, but more quickly and efficiently. However, technology is not foolproof and there are cases where you can make wrong decisions. Having the correct controls in place upfront and ensuring that service providers can help will minimize the impact of these mistakes for the benefit of good deeds.

12 Conclusions and Future Work

Through the development of this work it has been possible to verify the great importance of the representation learning framework, it is truly powerful, the framework was consistent across the different tests that were performed. As expected, the ”contamination” hyperparameter was one of the most important, since it depends on how restrictive or how free the model is when identifying fraud, therefore, the value of this hyperparameter must be chosen. through a good level of mastery of the topic, since in real life, you will not have labeled datasets. About feature engineering, we identify that there are certain patterns related to certain variables such as the channel, the amount of the transaction, even the day of the week, so it may be a good idea to use groupings or characterize behaviors of the variables according to its occurrence within the data set. Moreover, it was noted the importance of making a correct feature engineering, since erroneous information given by the features can lead to bad results. On the other hand. filtering and selecting the best features is a great importance, considering that some features can be ambiguous or not give enough information.

Regarding future work, although the implemented technique for this work was successful, it has a problem which is that it evaluates the characteristics globally, which limits the power of filtering, having said this, it could be thought of as future work to look for much more robust frameworks to filter than also take into account local characteristics, for example, Dynamic Combination of Detector Scores for Outlier (DCSO), see Zhao & Hryniewicki (2019). Also, it is important to mention and propose for the future an improvement regarding the variance problem that became evident in the cross-validation and is the possibility of implementing training routines in parallel such as bagging or opting for larger structures of trees, given that tree structures such as the Isolation Forest were implemented for this work, but a much larger Isolation Forest could be used.

Finally, it is key to highlight the importance that this project has had and will have for the training and intellectual enrichment of its authors because as the project developed, the concepts learned and the lessons taught by professors Tomás Olarte and Santiago Hernández during the semester were extremely useful to solve various problems that arose through the completion of it, this attests to its importance when transferring the content of the classes to projects with a vision that can serve our society in the future.

References

Alelyani, Salem, Tang, Jiliang, & Liu, Huan. 2018. Feature selection for clustering: A review. *Data Clustering*, 29–60.

Boukerche, Azzedine, Zheng, Lining, & Alfandi, Omar. 2020. Outlier detection: Methods, models, and classification. *ACM Computing Surveys (CSUR)*, **53**(3), 1–37.

Domingues, Rémi, Filippone, Maurizio, Michiardi, Pietro, & Zouaoui, Jihane. 2018. A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognition*, **74**, 406–421.

Liu, Fei Tony, Ting, Kai Ming, & Zhou, Zhi-Hua. 2008. Isolation forest. 413–422.

Meng, Fanrong, Yuan, Guan, Lv, Shaoqian, Wang, Zhixiao, & Xia, Shixiong. 2019. An overview on trajectory outlier detection. *Artificial Intelligence Review*, **52**(4), 2437–2456.

Perols, Johan. 2011. Financial statement fraud detection: An analysis of statistical and machine learning algorithms. *Auditing: A Journal of Practice & Theory*, **30**(2), 19–50.

Sudjianto, Agus, Nair, Sheela, Yuan, Ming, Zhang, Aijun, Kern, Daniel, & Cela-Díaz, Fernando. 2010. Statistical methods for fighting financial crimes. *Technometrics*, **52**(1), 5–19.

Walker, John. 1999. How big is global money laundering? *Journal of Money Laundering Control*.

Zhao, Yue, & Hryniewicki, Maciej K. 2018. XGBOD: improving supervised outlier detection with unsupervised representation learning. 1–8.

Zhao, Yue, & Hryniewicki, Maciej K. 2019. DCSO: dynamic combination of detector scores for outlier ensembles. *arXiv preprint arXiv:1911.10418*.

Zhao, Zheng, & Liu, Huan. 2007. Spectral feature selection for supervised and unsupervised learning. 1151–1157.

Zimek, Arthur, & Filzmoser, Peter. 2018. There and back again: Outlier detection between statistical reasoning and data mining algorithms. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, **8**(6), e1280.

Zimek, Arthur, Campello, Ricardo JGB, & Sander, Jörg. 2014. Ensembles for unsupervised outlier detection: challenges and research questions a position paper. *Acm Sigkdd Explorations Newsletter*, **15**(1), 11–22.