# UNIVERSITA' DEGLI STUDI DI PADOVA

## DIPARTIMENTO DI SCIENZE ECONOMICHE ED AZIENDALI

## "M. FANNO"

## CORSO DI LAUREA IN ECONOMIA

**PROVA FINALE**

# "Machine Learning and Econometrics in Credit Card Fraud Detection: An Empirical Analysis"

**RELATORE:**

**CH.MO PROF. Luca Nunziata**

**LAUREANDO: Mattia Marzaro**

**MATRICOLA N. 1190655**

**ANNO ACCADEMICO 2022 – 2023**

Dichiaro di aver preso visione del “Regolamento antiplagio” approvato dal Consiglio del Dipartimento di Scienze Economiche e Aziendali e, consapevole delle conseguenze derivanti da dichiarazioni mendaci, dichiaro che il presente lavoro non è già stato sottoposto, in tutto o in parte, per il conseguimento di un titolo accademico in altre Università italiane o straniere. Dichiaro inoltre che tutte le fonti utilizzate per la realizzazione del presente lavoro, inclusi i materiali digitali, sono state correttamente citate nel corpo del testo e nella sezione ‘Riferimenti bibliografici’.

*I hereby declare that I have read and understood the “Anti-plagiarism rules and regulations” approved by the Council of the Department of Economics and Management and I am aware of the consequences of making false statements. I declare that this piece of work has not been previously submitted – either fully or partially – for fulfilling the requirements of an academic degree, whether in Italy or abroad. Furthermore, I declare that the references used for this work – including the digital materials – have been appropriately cited and acknowledged in the text and in the section ‘References’.*

Firma (signature) ….Mattia Mutaro…………………

# Abstract

The exponential expansion in the volume of online transactions in the last years has evidenced how the underlying risk of credit card fraud is also rising. Addressing this topic, a potential solution can be found within the use of new computational technologies, such as machine learning algorithms, combined with econometric analysis tools.

The purpose of this work is building a Machine Learning model in order to be able to detect credit card fraud with the highest degree of accuracy possible, starting from the analysis of a dataset created with PaySim, a software that simulates credit card transactions based on a real dataset, due to privacy reasons.

The empirical analysis was performed through diverse machine learning algorithms, resulting in the selection of a specific type of Random Forest algorithm, the Balanced Random Forest (BRT) algorithm. The selection of this specific algorithm was due to its elevated accuracy in both model building and in handling the severe class imbalance issue that emerged in the analysis of the dataset.

Resulting from the analysis and model building it is safe to say that utilising new technologies as machine learning algorithms can bring a great advantage in detecting credit card fraud cases and substantially improve protection from this kind of threat that is everyday more relevant due to the progressive digitalization of our economic system.

# Abstract - Italiano

La crescita esponenziale del volume delle transazioni online negli ultimi anni ha evidenziato come anche il rischio di frodi su carte di credito stia anch'esso aumentando. Una potenziale soluzione a questo problema può essere ricercata nelle nuove tecnologie come gli algoritmi di machine learning combinati con strumenti di analisi econometrica.

Lo scopo di questo studio è la costruzione di un modello di machine learning che rilevi le frodi su carte di credito con il più alto grado di accuratezza possibile, partendo dall'analisi di un dataset creato con PaySim, un software che simula transazioni di carte di credito basandosi su un dataset reale, per motivi di privacy.

L'analisi empirica è stata perseguita tramite diversi tipi di algoritmi di machine learning, con la selezione di uno specifico tipo di algoritmo, il Balanced Random Forest (BRT). La scelta di questo specifico tipo di algoritmo è dovuta sia all'accuratezza del modello risultante che alla capacità di affrontare il problema di grave sbilanciamento delle classi emerso dall'analisi del dataset.

In seguito all'analisi ed alla costruzione del modello si può affermare che l'utilizzo delle nuove tecnologie, quali il machine learning, può generare un grande vantaggio nel rilevamento di casi di frodi su carte di credito  e sostanzialmente migliorare la protezione da questo tipo di minacce che è ogni giorno più rilevante data la progressiva digitalizzazione del nostro sistema economico.

# 1. Introduction

There is no doubt that our economic system has radically changed in the last twenty years, in large part due to great technological innovations and that digitalization has had a central role in this transition, granting a globally interconnected system, with transactions taking place at unprecedented  extremely fast rates. Among these great changes, in our ever more digital economy, this study focuses its attention on the exponential growth of credit card transactions and its consequences.

As stated in one of the latest studies by the U.S. based Federal Reserve (FEDERAL RESERVE, 2022. *Federal Reserve 2022 Payments Study*) the total number of non-cash payments in the U.S. grew in the period between 2018 and 2021 with the fastest rate since 2000. Focusing in depth on data provided by the FRPS, it is noticeable how the number of transactions on credit cards is on a growing trend (*chart 1.1*).

At the same time, with the increase in volume and frequency of transactions, some downsides can be found, the main being the increase in numbers of the cases of credit card fraud. This kind of issue is a natural consequence to the increase in the numbers of transactions, but this is not a reason for it to be underestimated. Another report from the Federal Reserve, dated 2016 and last redacted in 2018 (FEDERAL RESERVE, 2018. *Changes in U.S. Payments Fraud from 2012 to 2016: Evidence from the Federal Reserve Payments Study*), evidence shows how the number of credit card fraud cases in the U.S. is also rising, with data referring to the 2012-2015 period. In *chart 1.2* it can be seen how the total volume of fraud between 2012 and 2015 went from 6,1 to 8,34 billions of dollars. Evidence reports  how  the problem is not only limited to the U.S, the European Central Bank also released a report in 2022 (EUROPE CENTRAL BANK, 2022. *Seventh report on card fraud*) where proofs are provided of the fact that credit card fraud cases are rising in the Eurozone within the SEPA (Single Euro Payment Area) system. In the study it is specified how fraudulent transactions are constantly growing, while on the other hand the volume of  overall transactions is growing faster, thus partially hiding the issue in data.

In order to deal with the Credit Card Fraud issue, the aim of this study is to adopt an econometric approach, exploiting new technologies, specifically machine learning algorithms to see if it's possible to mitigate the problem, detecting Credit Card Fraud and eventually protecting the consumers from this serious financial menace. While Machine Learning is defined by many as a branch of Artificial Intelligence, and it grew a separated culture with respect to econometrics, (substantially ending up being considered as different fields) these two fields share the common goal of building models for the prediction of determined event outcomes. Specifically, in this dissertation, machine learning algorithms will be used only to predict economic phenomena, thus granting the possibility to relatively compare an econometric approach to the use of a machine learning model and its evaluation to predict, or in this case detect, credit card fraud cases.

Following this introductory chapter, a brief literature review will be made, concerning the study materials used in order to finalise this research, while trying to discuss previous findings, eventual problems found and lacking material in literature.

After the literature review the data source and reliability will be thoroughly analysed, with the explanation of the PaySim software through which the database was produced. In that same chapter pre-processing of data will be treated, to introduce the database contents and contextualise them.

Then the fourth chapter regarding methodology will be presented, with the explanation of how the final machine learning model was produced and where the rationale behind it will be exposed. During the same chapter the training process and prediction phase of the model will be discussed, with an eventual discussion of the potential problems arisen in the process.

The fifth chapter will contain the evaluation metrics utilised in order to assess the model performance, comparing the different results obtained through the change in variables during the training process.

Then the sixth and last chapter will then be utilised to discuss the findings from the model, addressing the implications of the model output. Lastly some personal conclusions will be traced, to summarise the dissertation and possibly find if the tools of machine learning could be used to improve security in the credit card transactions field.

# 2. Literature review

The field of credit card fraud is receiving rising attention in the last few years, due to the fact that online transactions are exponentially rising, as already stated in the introduction, but in truth the research on this topic is not so recent. One of the oldest studies found on credit card fraud and a potential solution for its detection dates back to 1994, with a paper by Ghosh and Reilly (GHOSH, S. and REILLY, D. L., 1994. *Credit Card Fraud Detection with a Neural-Network, 1994 Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences,* Wailea, HI, USA, 1994, doi: 10.1109/HICSS.1994.323314). In their paper the researchers trained a neural network to detect credit card fraud, and even if the types of fraud in 1994 may have been very different from the ones that are nowadays perpetrated, this study is probably one of the earliest in the field and can be considered as the foundation for the newest research articles.

Another paper by Chaudhary, Yadav and Mallick, written in 2012 and published in the International Journal of Computer Applications (CHAUDHARY, K., YADAV, J. and MALLICK, B., 2012. *A review of Fraud Detection Techniques: Credit Card,* International Journal of Computer Applications, vol. 45, no. 1), can be considered of great relevance to the credit card fraud study. In their paper the authors delineate the different types of fraud (Credit card fraud, Bankruptcy fraud, Counterfeit fraud, etc.) and then proceed to explain the possible tools to solve the problem, such as neural networks (citing Ghosh and Reilly), decision trees (which is the method that will be adopted in this dissertation), logistic regression and other techniques. The paper in itself does not contain any experiments on datasets, but gives general information and insights on the topic, bringing attention to the matter of credit card fraud.

Moving on to more experimental research, there is a paper from Xuan et al. (XUAN, S., et al., 2018. *Random forest for credit card fraud detection*, 2018. *IEEE 15th International Conference on Networking, Sensing and Control (ICNSC)*, Zhuhai, China, 2018, doi: 10.1109/ICNSC.2018.8361343) published in 2018 by the Institute of Electrical and Electronics Engineers (IEEE), that utilises a dataset obtained from a Chinese ecommerce provider to test some models on credit card fraud detection. In the paper the authors created different Random Forest models for the detection of fraud cases, but as the authors themselves mention, the models perform better on a restricted subset, because on the original dataset the fraud ratio is 0,27%. This ratio causes a severe problem of class imbalance, greatly reducing the accuracy of the model. To obviate this problems the authors utilise in one of their three experiments a random undersampling technique (undersampling, oversampling and class imbalance will be further discussed in chapter 4 of the dissertation, while debating the methodology applied to this research's machine learning model), although in the conclusions the authors admit that class imbalance is still problematic, the results are not fully satisfactory and that the models could perform better on smaller and more balanced datasets.

Another similar paper was also published by IEEE in 2019, by Suresh Kumar et. al (SURESH KUMAR, M., et al., *Credit card fraud using random forest algorithm*, 2019. *2019 3rd International Conference on Computing and Communications Technologies (ICCCT),* doi: 10.1109/ICCCT2.2019.8824930), where the authors apply a random forest algorithm to a real world dataset provided by a unspecified "large European company" obtaining an accuracy of about 90%.

Given the said papers and an abundance of other articles available on the topic, it is possible to notice how there are still some problems in credit card fraud detection due to possible class imbalances that could arise from large datasets and the general literature around these cases seems to be lacking of material. This paper then analyses a specific case where this problem is present, treating a large dataset with a severe problem of class imbalance, as it will be further specified during the next chapter, while trying to find different solutions in alternative algorithms.

Apart from these articles the general study about the basis of the topic, specifically machine learning algorithms and techniques, were done through the reading of two books, these being *An Introduction to Statistical Learning with Applications in R* (JAMES, G., WITTEN, D., HASTIE, T. and TIBSHIRANI, R., 2021. *An Introduction to Statistical Learning with Applications in R,* 2nd edition) and *The Elements of Statistical Learning Data Mining, Inference, and Prediction* (HASTIE, T., TIBSHIRANI, R. and FRIEDMAN, J., 2009. *The Elements of Statistical Learning, Data Mining, Inference, and Prediction*, 2nd edition).

This being said, it is now time to deep dive into the dataset itself, with the motivation behind the choice of it and its peculiar nature that will be explained in the next chapter.

# 3. Data source and pre-processing

When dealing with a machine learning model the first and foremost requirement needed to the training process in order to be effective is a large enough and properly pre-processed dataset. This is a thing not to underestimate, especially in the case of this field of research, due to the fact that treating financial transactions data is a delicate subject in which precision and efficiency are necessary.

Financial transactions are by their nature private, and not available to the public. This arises a problem in this research on credit card fraud and generally on research around other types of financial data. Datasets containing financial transactions could theoretically be deprived of personal data and eventually be accessed through some special permits given to researchers from private or public entities owning these datasets, but this could bring significant resource losses in terms of time and eventually on the economical side.

A potential and useful solution to this problem is provided by a study by Lopez-Rojas, Elmir and Axelsson (LOPEZ-ROJAS, E. A., ELMIR, A., AXELSSON, S., 2016. *PAYSIM: A FINANCIAL MOBILE MONEY SIMULATOR FOR FRAUD DETECTION*, Proceedings of the European Modeling and Simulation Symposium) in which the authors produced a software, PaySim, to obviate to the issue of financial data privacy.

PaySim is a software based on Agent-Based simulation techniques that generates large datasets of transactions, starting from a real world dataset provided by a non-disclosed multinational company from an African Country. This allows researchers to gain access to datasets that have high reliability in terms of data and are very close to a real world situation, therefore granting good quality data for training a model.

Before entering the specific case of this dataset it is necessary to introduce the programming language that was utilised in this research to create the model. This language is R, a powerful language created specifically for statistical and econometric analysis (https://www.r-project.org/) and which will be the language present in the code snippets that will be provided in the body of the dissertation and in the appendix.

In the specific case of this research the database selected was already created and made available through the Kaggle platform (the link to the dataset will be provided in the appendix). The dataset is composed of around 6.3 million observations with 11 variables for each observation, with said variables being for example the type of the transaction (cash-in, cash-out, payment, transfer, etc.), the amount of the transaction, the identifiers of the two parts of the transaction, the *isFraud* variable, which will be the focus predictor of the model and another variable that defines if the transaction was detected as Fraud or not before the application of the machine learning model (namely the *isFlaggedFraud* variable). The *isFraud* is a binary variable that assumes value 1 when the observation is a fraudulent case and value 0 for a legitimate transaction.

The high importance of this variable is due to the fact it defines that the database already has labelled the fraudulent transactions, and determines this case as a supervised learning situation, in which labelled data are used to train the model to recognise fraudulent cases based on patterns in data.

During the observation of the dataset contents and of the *isFraud* variable a noticeable fact emerged, as was precedently introduced during the previous chapter, that a severe issue of class imbalance was present. Having already introduced that the dataset is composed of 6.3 million observations, it is important to point out that just 8213 of these transactions are fraudulent, resulting in a percentage of a mere 0.13%. Such a case is an interesting one for research, since as previously stated there is a problem in credit card fraud detection literature with datasets having class imbalance issues and this research has tried a specific algorithm, which will be discussed in next chapter, to try to resolve this issue.

Starting with the pre-processing process of the dataset, needed to clean and render the dataset more efficient for the model to be trained on, some variables that were unnecessary to the training were removed, specifically the variables about the numerical identifiers of the transaction, the transaction type alphanumeric value, the identifiers of the transaction parts and the already mentioned *isFlaggedFraud* variable, which is not necessary for model training. In the appendix of the dissertation the code of this and following pre-processing steps will be present.

Another issue is that a high number of observations could be very taxing in terms of computing resources and time for the training process, so after careful consideration the dataset was reduced to include a number of 2 million observations. While scaling down the dataset the *isFraud* variable maintained closely the same ratio between positive and negative outcomes, passing from 0.13% c.ca to 0.10% for the positive case, so the class imbalance issue is not significantly different from when considering the whole dataset. The last step in the preprocessing of the dataset was splitting the dataset into two subsets, one utilised for the training process and the other for testing the model, specifically the dataset was split so that the training subset is taking randomly half of the observations, while the test subset takes the other half, as for common practise in model training.

# 4. Methodology, training process and predictions

After careful consideration of already existing literature and the study of theory, the chosen approach to detect credit card fraud in this dissertation was chosen to be the Random Forest algorithm. Random Forest algorithm is a technique originated from the wider family of the decision tree algorithms, that can be utilised both for regression and classification problems. In the case of credit card fraud detection the problem is a case of classification, specifically of predicting if an observation can be classified as fraud or not with a binary variable. In order to better explain how a Random Forest algorithm works it is appropriate to first address what is a decision tree.

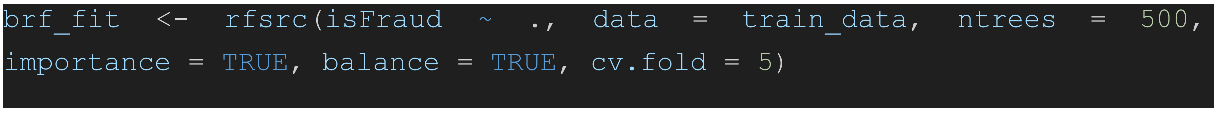
A decision tree is a structure starting from a node, the *root node*, that sequentially splits into other internal nodes (connected by so called branches) based on a decision taken on a specific feature (for example in the case of credit card fraud if the transaction amount is over or under a specific value). Based on the algorithm a tree can have a high or low amount of nodes and branches depending on how many different feature decisions are made. At the end of the tree, which is often visually represented from top to bottom with the root at the top, are the final nodes, called leaves, that represent the different outcomes of the tree, that in the case of this dissertation will be if the single case will be classified as fraud or not. To decide which the outcome will be, the single observation passes through the various decisional nodes and based on its features the decision will result in a different termination leaf (outcome).

The Random Forest algorithm is called an ensemble method, due to the fact that it combines different decision trees in order to boost its predictive accuracy and potentially avoiding overfitting of the model, caused by a single decision tree that could possess an unnecessary complex structure of nodes that causes overfitting problems. Differently from the single decision tree, the Random Forest algorithm is, as said, composed of a high number of different trees, but far less complex in their singular structure. To obtain a reliable prediction, the algorithm utilises a technique called bootstrapping sampling, that creates multiple subsets from the training dataset and trains each single tree with a different (and randomised) subsample. It is also important to note that every tree in the model has a random feature selection at each node, granting the model some randomness and reducing the risk of overfitting. Lastly, in the decision process and in the case of classification trees, each tree ends up predicting an outcome from its input, and the final outcome of all trees is decided through the majority vote of the outcome class (the most present outcome between all trees will be the selected one).

While this kind of algorithm performs with really good accuracy for classification problems like the one in this dissertation, it is also true that, as stated in other literature precedently discussed, this kind of algorithm has some problems with imbalanced datasets. The solution to this issue was found through a special type of Random Forest algorithm called Balanced Random Forest (BRF), that can be found in the *RandomForestSRC* package in R. The BRF algorithm has its foundations in Random Forest algorithm, while having some key differences in order to address imbalanced datasets as the one provided in this study.

Starting from the subsets created at the start of the process (bootstrap sampling), the BRF algorithm aims to obtain balanced subsets to train on the different trees; this is done through a mix of oversampling of the minority class and the undersampling of the majority class. The undersampling of the majority class (in this case the event of having a legitimate transaction) consists in the random selection of some instances of the majority class in order to reduce the presence of it in the subset, while the oversampling of the minority class is obtained through the random oversampling technique (and not with other techniques such as SMOTE or ADASYN), consisting in creating copies of random instances of the minority class (fraudulent transactions). This mix of oversampling and undersampling creates more balanced subsets with which the trees can be fitted (trained). Another key difference between this algorithm and the standard random forest is found in the decision process, since the BRF applies a more probabilistic approach, with the outcomes being class probabilities instead of positive and negative values only resulting then in an outcome value between 0 and 1.The probabilities of every tree are calculated and a successive step is required, with the establishment of a threshold value, over which the outcome is assigned to the positive class and is assigned to the negative class in the case of being under the same threshold value.

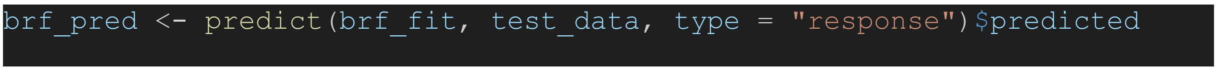
Proceeding with the training process of the model, the code snippet of the model training will now be provided and explained in order to clarify in detail how it was done.



```
brf_fit <- rfsrc(isFraud ~ ., data = train_data, ntrees = 500,
importance = TRUE, balance = TRUE, cv.fold = 5)
```

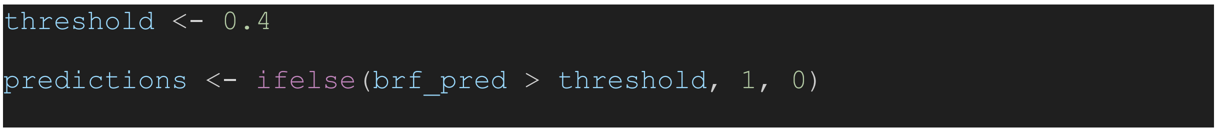
The *brf_fit* is the variable storing the model, which is then trained through the *rfsrc* function from the *RandomForestSRC* package. Then the target predictor (in this case the *isFraud* variable) and the subset in which to train the model on, namely the *train_ data* subset created during the pre-processing phase, are specified. After several tries the optimal number of trees to use was found to be 500, with the overall accuracy of the model worsening with different values; during next chapters while discussing the outputs of the model it will be possible to see this in detail. The optional command *importance = TRUE* specifies the model to evaluate how impactful the different variables in the dataset are to the training of the model, to potentially provide insights on how much influence each variable has during the prediction phase. The focus of the code snippet is the *balance = TRUE* command, which specifies that the BRF model will be used, in order to handle data imbalance as previously discussed. The last command explains how an additional measure was utilised in order to obtain even a more robust model to improve the accuracy and further lessen the impact of the class imbalance issue, through the technique of k-fold cross-validation, with the number of 5 folds that was found to be optimal to improving the model accuracy. The k-fold cross validation consists in this case of an ulterior process of dividing the dataset into 5 subsets (the so called folds) and iterating the training model process 5 times, with every time taking one of the folds as the validation subset and utilising the other 4 for the training. After this iteration process is repeated for each fold (so for 5 times) the results of the model are averaged to provide an even better and more accurate model.

After the training process is complete the following step is to utilise the model to predict the potential fraud cases via the *test_data* subset created during the pre-processing phase (half of the dataset). The training code is the following:



```
brf_pred <- predict(brf_fit, test_data, type = "response")$predicted
```

With this command, values are predicted utilising the model precedently trained *brf_fit*, over the testing subset *test_data.* The other commands present in the code simply functions as a way to calculate and store the prediction class probabilities values. After the predictions are calculated and stored in the b*rf_pred* variable there is the need, as precedently said, to convert these class probabilities into binary values (0 or 1) through the threshold value. During the study various threshold levels were attempted, with the value of 0.4 that was found to be the optimal one for the accuracy of the model to be maximised (the various levels of thresholds utilised and its outputs will be discussed in the next chapter). The code utilised to do so is the following:



```
threshold <- 0.4

predictions <- ifelse(brf_pred > threshold, 1, 0)
```

In the next chapter the model prediction outputs and its evaluation metrics will be assessed.

# 5. Model evaluation metrics

During the experiment the model was trained several times with different parameters in terms of threshold value and number of trees in order to find whichever was the optimal parameters set to maximise the effectiveness of the model. Talking about the effectiveness of the model implies that the performance of it was assessed through techniques of model evaluation.

The main instrument used to evaluate the model is the confusion matrix, a tool providing different measures of performance, such as the overall accuracy of the model, the number of true and false positives, true and false negatives, and other indexes. The code used for the confusion matrix is really simple to implement and is the following:

```
confusion_matrix <- confusionMatrix(predictions, target, positive =
"1")
```

Here the confusion matrix has to be given as input the value of the predictions and the target predictor variable, that was precedently specified in the code (*isFraud* variable) and the positive class is to be specified being the class having value 1 (the case of a fraudulent transaction) The output of the matrix can then be seen to evaluate the model.

The fact that the dataset has a problem of class imbalance still reflects even in the performance assessment, where the overall accuracy given as output in the confusion matrix is not a reliable evaluation parameter, because it is calculated as follows:

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Number\ of\ Predictions}$$

This calculation is not representative of the true model accuracy because of the high number of non fraudulent cases in the dataset, which results in an overestimation of the accuracy of the model. In order to better assess the performance another value present in the confusion matrix can be used, the so called *Balanced Accuracy*, that is instead calculated as:

$$Balanced\ Accuracy = \frac{1}{2}(Sensitivity + Specificity)$$

The *Balanced Accuracy* is then a mean of sensitivity and specificity of the two classes, with the *Sensitivity* being the true positive ratio and the *Specificity* being the true negative ratio for the negative class. Considering this specification on sensitivity the equation could be rewritten as:

$$Balanced\ Accuracy = \frac{1}{2}(\frac{True\ Positives}{True\ Positives + False\ Negatives} + \frac{True\ Negatives}{True\ Negative + False\ Positives})$$

To have an even clearer idea on how the model is performing an ulterior performance metric was calculated, the F1 score, another performance assessment tool utilised in assessing imbalanced datasets, that calculates the capability of a model to correctly predict positive cases. The F1 score is calculated as follows:

$$F1 = \frac{2 \times Precision \times Sensitivity}{Precision + Sensitivity}$$

*Precision,* also called Positive Predictive Value, is instead calculated as:

$$Precision = \frac{True\ positives}{True\ Positives + False\ Positives}$$

While the Balanced accuracy focuses on the effectiveness with which the model assesses both of the classes, the F1 score gives more focus on correctly predicting the positive class (the fraudulent cases). This could be useful in the case of a model needing to be more accurate in predicting a class while not having a real reason to focus on the negative class. In the case of this study, the model was found to be performing the best with a set of parameters that also results in both the best balanced accuracy and f1 score. That could also not always be true and a trade-off between predicting more accurately the positive class while predicting with inferior accuracy the negative class (or the contrary, depending on the choice) should have been undertaken.

The Table in *chart 5.1* summarises the main differences between the model performances resulting from the training with different parameters, specifically the number of trees and threshold value. As previously mentioned the overall accuracy seems to be biassed due to class imbalance, while the balanced accuracy and F1 score yield a more realistic and accurate result for the model accuracy. Regarding the statement, during the previous chapter, about the optimal number of trees being found to be 500, it is now noticeable how this was discovered during the experiment, with the specific case of the number of trees equal to 500 and the threshold value set to 0.4 being the best solution in order to maximise both the model accuracy and F1 score.

# 6. Results discussion and conclusion

With the previously discussed experiment this dissertation aimed to find if machine learning could be a good tool for credit card fraud detection, even in the case of having an highly imbalanced dataset. Through the output of  the confusion matrix it can be seen how the model trained with the Balanced Random Forest algorithm seems to be a reliable way to detect fraudulent transactions, but even so the model is not perfect.

In the best version of the model in terms of performance, with 0.4 as the threshold value and 500 trees to train it, there was still a small but not non-existent portion of observations that were misclassified by the algorithm, specifically with 45 false positives and 405 false negatives.

Effectively the total number of misclassified cases are just 450 over a testing dataset of 1 million instances, but while this could be considered a insignificant number of misclassification, in a real world scenario even a small number of fraud cases could potentially bring a great harm to both consumers and firms.

It is true, on the other hand, that realistically a model considering such a high number of transactions can't possibly have perfect results in its classification, and that in respect to other kinds of simpler models and even classical Random Forest models, this approach yields far better results.

Some possible alternatives as machine learning algorithms and techniques not treated in this dissertation could be Support Vector Machines (SVM), neural networks or alternatively some unsupervised learning models (for example with the K-means clustering algorithm). These models could provide another approach to the problem and possibly improve further the detection capability.

In conclusion it is safe to say that machine learning approaches in general are a good method in aiming to provide protection from credit card fraud cases. In cases such as one with a highly imbalanced dataset, plausible to happen in a real world scenario, algorithms as the one utilised in this study, are reliable and have a good performance, even if some ulterior improvements could be made.

# 7. References

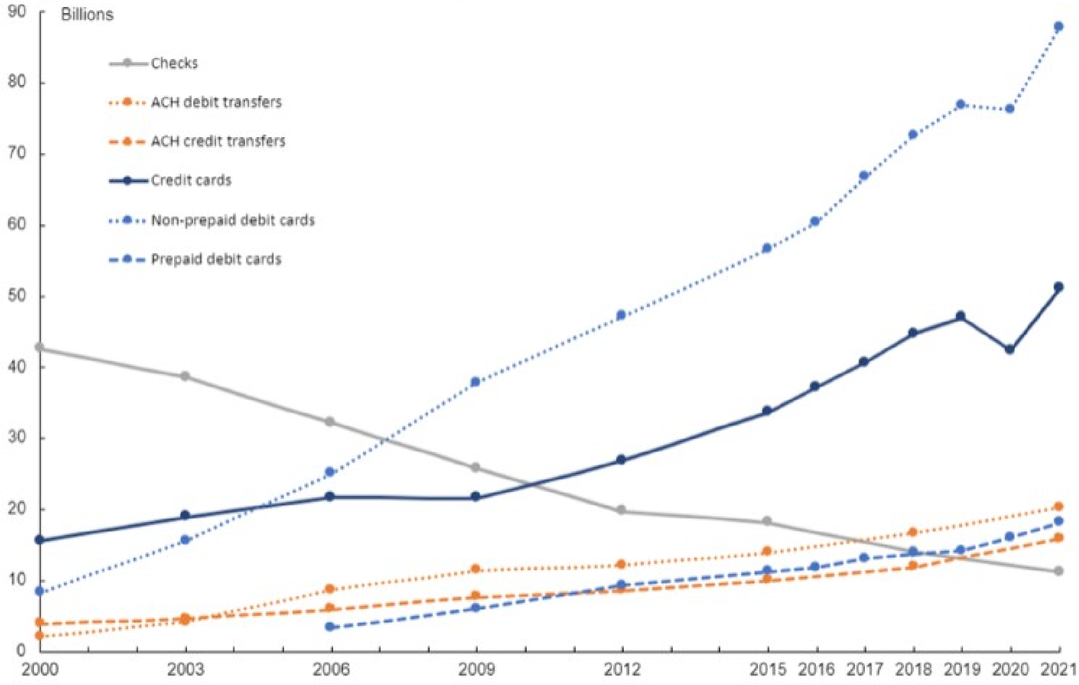
- FEDERAL RESERVE, 2022. *Federal Reserve 2022 Payments Study*
- FEDERAL RESERVE, 2018. *Changes in U.S. Payments Fraud from 2012 to 2016: Evidence from the Federal Reserve Payments Study.*
- *EUROPE CENTRAL BANK, 2022. Seventh report on card fraud.*
- *GHOSH, S. and REILLY, D. L., 1994. Credit Card Fraud Detection with a Neural-Network, 1994 Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences, Wailea, HI, USA, 1994, doi: 10.1109/HICSS.1994.323314.*
- *CHAUDHARY, K., YADAV, J. and MALLICK, B., 2012. A review of Fraud Detection Techniques: Credit Card, International Journal of Computer Applications, vol. 45, no. 1.*
- *XUAN, S., et al., 2018. Random forest for credit card fraud detection, 2018. IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), Zhuhai, China, 2018, doi: 10.1109/ICNSC.2018.8361343.*
- *SURESH KUMARr, M., et al., Credit card fraud using random forest algorithm, 2019. 2019 3rd International Conference on Computing and Communications Technologies (ICCCT), doi: 10.1109/ICCCT2.2019.8824930*
- *LOPEZ-ROJAS, E. A., ELMIR, A., AXELSSON, S., 2016. PAYSIM: A FINANCIAL MOBILE MONEY SIMULATOR FOR FRAUD DETECTION, Proceedings of the European Modeling and Simulation Symposium*
- *https://www.r-project.org/*
- *https://www.randomforestsrc.org/*

# 8. Appendix

## Charts

### 1.1) Trends in non cash payments

Figure 2. Trends in noncash payments, by number, 2000–21



FEDERAL RESERVE, 2022. *Federal Reserve 2022 Payments Study*

1.2) Payments Fraud

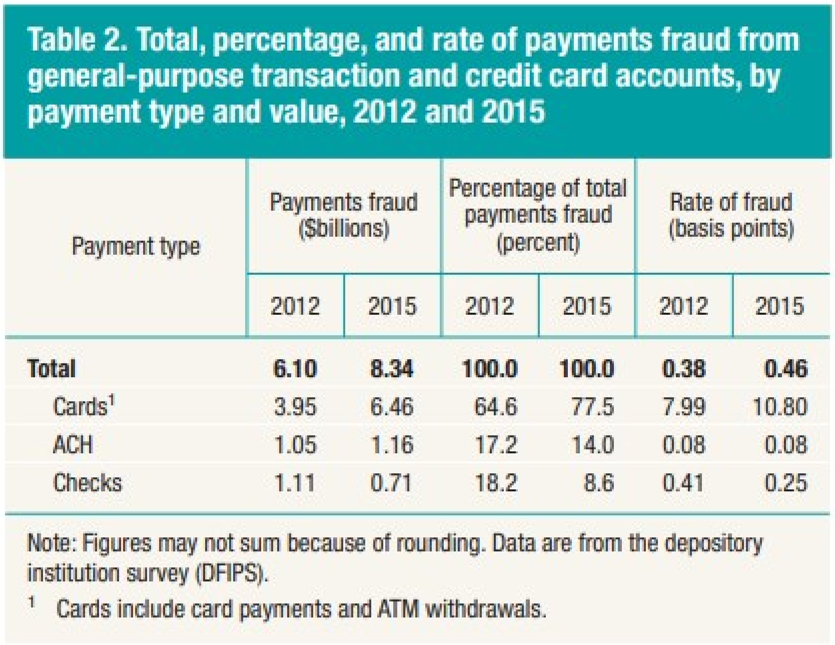


**Table 2. Total, percentage, and rate of payments fraud from general-purpose transaction and credit card accounts, by payment type and value, 2012 and 2015**

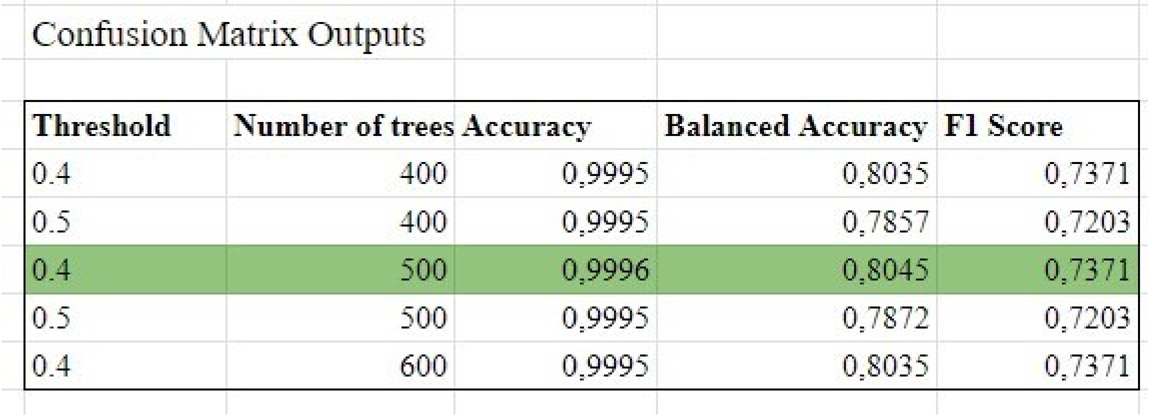
| Payment type | Payments fraud ($billions) | | Percentage of total payments fraud (percent) | | Rate of fraud (basis points) | |
|---|---|---|---|---|---|---|
| | 2012 | 2015 | 2012 | 2015 | 2012 | 2015 |
| **Total** | **6.10** | **8.34** | **100.0** | **100.0** | **0.38** | **0.46** |
| Cards[1] | 3.95 | 6.46 | 64.6 | 77.5 | 7.99 | 10.80 |
| ACH | 1.05 | 1.16 | 17.2 | 14.0 | 0.08 | 0.08 |
| Checks | 1.11 | 0.71 | 18.2 | 8.6 | 0.41 | 0.25 |

Note: Figures may not sum because of rounding. Data are from the depository institution survey (DFIPS).

[1] Cards include card payments and ATM withdrawals.

FEDERAL RESERVE, 2018. *Changes in U.S. Payments Fraud from 2012 to 2016: Evidence from the Federal Reserve Payments Study*

5.1) Confusion Matrix Outputs with different parameters sets



Confusion Matrix Outputs

| Threshold | Number of trees | Accuracy | Balanced Accuracy | F1 Score |
|---|---|---|---|---|
| 0.4 | 400 | 0,9995 | 0,8035 | 0,7371 |
| 0.5 | 400 | 0,9995 | 0,7857 | 0,7203 |
| 0.4 | 500 | 0,9996 | 0,8045 | 0,7371 |
| 0.5 | 500 | 0,9995 | 0,7872 | 0,7203 |
| 0.4 | 600 | 0,9995 | 0,8035 | 0,7371 |

## Dataset

The dataset utilised in this dissertation is available at:

https://www.kaggle.com/datasets/vardhansiramdasu/fraudulent-transactions-prediction/data

Model Code in R

```r
library(randomForestSRC)

library(caret)


# Reading the dataset and creating a sample to speed up computing, removed unnecessary columns

print("Loading the dataset, please wait...")

data <- read.csv("C:/Users/…/fraud_detection_dataset.csv", header = TRUE)

dataset <- data[, -c(1, 2, 4, 7, 11)]

dataset <- dataset[1:2000000, ]


# Setting seed to obtain replicability and creating training and testing subsets

print("Partitioning the dataset...")

set.seed(123)

train_indices <- createDataPartition(dataset$isFraud, p = 0.4, list = FALSE)

train_data <- dataset[train_indices, ]

test_data <- dataset[-train_indices, ]


# Fitting the Balanced Random Forest (BRF) model

print("Now fitting the model, please wait...")

brf_fit <- rfsrc(isFraud ~ ., data = train_data, ntrees = 500, importance = TRUE, balance = TRUE, cv.fold = 5)


# Making predictions on the test subset

print("Now making predictions based on the model trained...")

brf_pred <- predict(brf_fit, test_data, type = "response")$predicted


# Convert predictions and target to factors to calculate the confusion matrix

threshold <- 0.4

predictions <- ifelse(brf_pred > threshold, 1, 0)
predictions <- as.factor(predictions)
target <- as.factor(test_data$isFraud)


# Evaluating model accuracy on the test data with confusion matrix, positive class specified as 1
print("Evaluating Model accuracy...")
confusion_matrix <- confusionMatrix(predictions, target, positive = "1")
print(confusion_matrix)


#Further evaluation through the f1 score
precision <- confusion_matrix$byClass["Pos Pred Value"]
sensitivity <- confusion_matrix$byClass["Sensitivity"]
f1_score <- 2 * (precision * sensitivity) / (precision + sensitivity)
cat("F1 score is: ", f1_score)
```