POLYTECHNIC OF TURIN

Master's Degree in INGEGNERIA INFORMATICA (COMPUTER ENGINEERING)

Master's Degree Thesis

FORECASTING THE FINANCIAL RISK USING TIME SERIES ANALYSIS

Supervisors

Prof. Francesco VACCARINO

Prof. Luca CAGLIERO

Candidate

Mauro BELLINAZZI

July 2022

Abstract

The assessment of financial credit risk is a challenging and important research topic in the area of accounting and finance. Economic crises indicate that there is still no stable or globally valid solution for estimating the financial credit risk with sufficient accuracy. At the economic and banking level, credit institutions and credit information systems are looking for new methods of analysis on the data in their possession; in particular, the enormous amount of micro-transactions due to the advent of cashless transactions has not yet been exploited. In this dissertation, credit scoring models are proposed, using real payment data retrieved in a Payment Services PSD2 - Directive (EU) context. The data under analysis, i.e. a list of movements, is made available through the Account Information Service (AIS). But the data are not provided with a whole series of information to which credit bureaus and banks have access. Therefore, different solutions found in the literature have been extended and adapted to the available data. The goal of this thesis is to investigate the use of time series forecasting techniques to predict a credit score indicating the risk of incurring in a fraud from the history of payments recorded as bank transactions. This is particularly helpful to perform credit checks on customers who have no past credit history. In fact, the dataset under analysis are transactions made by users during a 3-month period.

In an initial supervised learning phase, we use a fraud label to train our model to classify users as fraudulent or not fraudulent. The fraud label was manually assigned by the company. Further details cannot be made publicly available for legal reasons. This approach is generalizable to the context we are seeing of open-Banking, where the fraud or non-fraud information would not be present.

In a second phase, Fraudolent and non-fraudolent users are divided into different cluster. All individuals (i.e all the time series) within the same cluster have the same spending behaviour. Upon confirmation by experts, the same credit score should be assigned to users belonging to the same cluster. The training of the algorithm cannot be supervised as it is not possible to rely on already calculated scores associated with the examples contained in the dataset.

In parallel, results obtained from time series forecasting are also added, allowing the time series to be extended into the future.

State-of-the-art research in banking risk management was compared also exploiting the additional fraud information to better control the clustering results. The

models most used by credit bureaus to analyze financial data have been analyzed and adapted in order to be usable on the dataset under analysis, and therefore more usable in the context of open banking that we will see in the coming years.

The results achieved and analyzed allow us to easily understand that there is no globally accepted method that has been shown to be better than others except on specific datasets that are not particularly significant at the level of structured research in this domain. Hence the importance of further analysis in this area by comparing different machine learning techniques to assess credit risk. In fact, we were able to obtain meaningful clusters by exploiting additional fraud information, but in a general context it would be appropriate to have financial domain experts able to validate and verify the cluster generation process on time.

Summary

The assessment of Financial Credit Risk is a challenging research topic involving governments, credit bureaus, banks, companies, private users and families.

The accuracy of Financial Credit Risk is of crucial importance to both the economy and society, in fact the risk analysis models of the institutes in charge of this evaluation have not proved to be sufficiently accurate. At the economic and banking level, credit institutions and credit information systems are looking for new methods of analysis on the data in their possession; in particular, the huge amount of micro-transactions due to the advent of cashless transactions has not yet been exploited.

The approach adopted today by lenders is based on traditional credit scores. Traditional or Bureau credit scores, e.g. Equifax, Experian and TransUnion use VantageScore or FICO Score, are proving inadequate and non-descriptive for individuals with no credit history.

The main criticisms of the current system are: the credit score changes only after values have been recorded, therefore only following a continuous and prolonged insolvency; income is often not considered, or is inaccurate and it is rarely possible to track where capital is invested; it is not possible to assign a credit score to those who have no credit history. In particular, they are unable to predict future spending capacities as they are based exclusively on the credit history and on the payment capacity, generally self-declaring, of consumers. The problem that is analyzed in this thesis is the possibility of associating a probability to each individual solely by observing bank transactions. This probability should therefore be representative of the risk of insolvency of the individual and therefore, in general, representative of his spending capacity, not just at a given moment in time, but sufficiently detailed to be able to describe its evolution over time and make predictions about future trends. The challenge is to create a new credit scoring system that can improve credit risk management by evolving with the user's financial history. This translates into being able to find a numerical estimate of the probability of solvency based solely on the individual's spending and earning behavior. To this purpose, the proposed approach adapts the credit score to consumer behavior by comparing it

with the credit score of other similar users.

In this dissertation, the proposed analysis methods models use real payment data retrieved in a PSD2 context. The PSD2 regulation is a European regulation that aims to make the management of money and payments safer and more convenient; among the various services it introduces the Account Information Service (AIS), which makes available the list of movements. These movements, as in the case of the dataset under analysis, are not provided with a whole series of information to which, on the other hand, credit institutions and banks have access. In the dataset under analysis, obtained through thesis collaboration with a company, transaction data are for example bank account transfers, electronic payments and/or credit/debit card payments, described by anonymous Ids (user, account/bank wallet and bank provider), two categorical labels (type of account and type of payment), amount and currency, and a Boolean label created by a human process of selection of fraud in SDD direct debit (Sepa Direct Debit).

The goal of this thesis is to investigate the use of time series forecasting techniques to predict a credit score indicating the risk of incurring in a fraud from the history of payments recorded as bank transactions. This is particularly helpful to perform credit checks on customers who have no past credit history. In fact, the dataset under analysis are transactions made by users during a 3-month period. This work reproduces the models used in the state of the art, on one hand using supervised machine learning methods to predict label fraud (Direct Debit Fraud), on the other hand using unsupervised clustering methods to categorize users into appropriate clusters. In addition, time-series forecasting on single user time series, was done to extract useful statistics. Future works should combine the results to generate as output a numerical value that is representative in terms of risk score and can provide a more appropriate tool for credit scoring analysis.

The biggest defect in the dataset is the lack of information about the direction of the transaction, i.e., it is not possible to determine exactly whether the transaction is incoming or outgoing and whether it has affected the balance. Another difficulty is the training of the algorithm, which cannot be supervised, i.e. it is not possible to rely on already calculated credit scores associated with the examples contained in the dataset. That is, the available inputs do not have a label that contains the numerical value that the algorithm is tasked with learning to predict.

The dataset under analysis contains for each transaction a set of labels, among which are a label user and a label fraud. The fraud label was manually assigned by the company during an independent research. Further details on the independent research cannot be made publicly available for legal reasons. By aggregating all

the transactions with the same label user, we create the user time series. A user is considered fraudulent because all his transactions were labeled as fraudulent, even though this could be not representative of the reality. In fact, in a real world scenario it is possible that a fraudulent user may also have non-fraudulent transactions.

In an initial supervised learning phase, we use the fraud label to train our model to classify users as fraudulent or not fraudulent. To make the supervised predictions, k neighbors time series classifiers were trained and compared. It must be noted that the dataset under analysis fraudulent users are about 0.01% of the total users. Given the strong unbalance of the data in this division, it is advisable to use special libraries of "unbalanced learning" able to increase the least represented class (in this case fraudulent) with mathematical techniques to improve learning.

In a second phase, Fraudolent and non-fraudolent users are divided into different cluster. We represent users with time series and we divide the users into cluster using k-means and k-shape unsupervised time series clustering methods. All individuals (i.e all the time series) within the same cluster have the same spending behaviour. Upon confirmation by experts, the same credit score should be assigned to users belonging to the same cluster.

Clustering methods allow one to use various metrics and approaches to divide the time series population into groups. Therefore, we investigated various metrics to assess the individual's situation and compare it to the financial situation of other individuals. To separate the data into clusters, we exploited the information generated by statistics analysis (min, max, avg, std, median, var, sum), and the metrics called "Dynamic Time Warping (DTW)" and "LB Keogh". In particular, it has proven useful to use DTW because it is a versatile algorithm for time series of different lengths to measure the similarity between them. However, its quadratic time and space complexity is an obstacle to its applications in large time series data mining and thus we used one of its lower-bound function: LB Keogh.

To differentiate and characterize individual users within the same cluster, the time series of individual users were also analyzed and partitioned using KMeans clustering. To select the appropriate number of clusters of each behavior, the number of clusters that reduced the reconstruction error was used. Using the clusters with lower reconstruction error it is possible to assume that these behaviors are significant, however they should be subjected to analysis by experts in finance and econometrics. In this step the best results (i.e. the minimum number of cluster needed to reconstruct the time series) were obtained by sorting the time series by their amount. Since we did not have domain expert opinion that could evaluate individual behaviors, it was only possible to use them for comparisons with similar behaviors with fraudulent or non-fraudulent users.

In addition to these methods, time-series forecasting on single user time series, was done to to extract useful statistics (seasonality, stationarity, trend, residual). These statistics will allow to distribute credit score over the entire time series of transactions. Two models were compared: Seasonal Autoregressive Integrated Moving-Average (SARIMA) and Prophet (by Facebook).

In particular, thanks to these libraries for the prediction of time series, by temporally dividing the user's time series into a train set and a test set, it is possible to calculate an error on the prediction and then use this information by properly evaluating this models; in fact, the results show that there are users who are "easy" to predict while others present more "unpredictable" transactions.

The current state of the art, allows us to easily understand that there is no globally accepted method that has been shown to be better than others except on specific datasets that are not particularly significant at the level of structured research in the field. Hence the importance of further analysis in this area by comparing different machine learning techniques to assess credit risk. The results obtained from the various calculations are not fully comparable with the results present in the state of the art as they do not contain the same information. In order to extract additional and financially meaningful information, further analysis by a domain expert would be required to add value to the clusters and predictions presented in this thesis. The same domain experts could validate and classify certain behaviors and/or users as virtuous based on economic analysis, in order to identify the set of users that most closely match those behaviors and propensities. Through these final steps, it would be possible to extend the methods analyzed by providing an additional tool for credit risk score analysis.

Acknowledgements

"A man who pays his bills on time is soon forgotten."
Oscar Wilde

The first ones I have to thank are certainly the professors of the Polytechnic of Turin who over the years have taught me the profession that I intend to practice throughout my career. A special thanks to all the professors in general who over the years have taught and collaborated in the courses I have taken. The main thanks goes to the professor who allowed the development of this thesis and therefore crowned the conclusion of my studies, Prof. Francesco Vaccarino. Immediately after surely comes the co-rapporteur, Prof. Luca Cagliero who allowed to set and focus the analysis on the most interesting parts of the project. At the same time I would also like to thank the company that hosted me together with all the colleagues and specialists in the field who listened to my questions and tried to answer all the questions and doubts. I would like to thank all those who have allowed me to write my thesis over the years and who have accompanied me through the years. Finally, I would like to thank everyone who has helped me not to give up and get this far and who has continued to believe in me despite everything. So I would like to thank my partner Silvia Zaccardi who encouraged me to finish my studies and move abroad. A very special thank you to my mother, Brunella Matarrese, and to my sister, Miriam Bellinazzi, who although far away geographically have always been there for me and have never let me lack anything.

Table of Contents

List of Tables

List of Figures

Listings

Chapter 1

Introduction

Problem statement

The problem we are going to analyse is the possibility of associating a risk probability to each individual solely by observing bank transactions (anonymized and therefore with a reduced level of useful information). This probability should therefore be representative of the risk of insolvency of the individual and therefore, in general, representative of his spending capacity, not just at a given moment in time, but sufficiently detailed to be able to describe its evolution over time and make predictions about future trends together with an error calculated on the data in our possession.

1.1 State of the art

In finance and banking, a financial risk score is defined as: *"The financial risk score measures the overall financial condition of a business or individual based on a number of credit measures that include typical items used in the credit evaluation process: UCC Deposits, Waivers, Outstanding Payments, etc. This measure is basically a provisional risk score for a fiscal entity, covering approximately the next 1 to 2 years. This compares to traditional scoring used by companies that evaluate and score business performance or that of their customers or partners."* [3]. In the academic literature and research conducted on Risk Scoring, along with the solutions to date in the banking, insurance and finance industries, we find it useful to introduce and cite the following publications that may be comprehensively descriptive of the state of the art.

1.1.1 revolution in risk management and compliance

This thesis and analysis are similar to those described in the work of B. van Liebergen, 2017 [4], the purpose of which is to shed light on the concept of machine learning and its uses within financial services. In particular, applications in the banking industry will be discussed through three use cases of machine learning: credit risk modeling, detection of fraud and money laundering, and surveillance of conduct breaches and abusive behavior within financial institutions.

		Linear methods	Non-linear methods
Problem type	**Supervised**		
	Regression	• Principal components • Ridge • Partial least squares • LASSO	Penalized regression: • LASSO • LARS • elastic nets Neural networks and deep learning
	Classification	Support vector machines	Decision trees: • classification trees • regression trees • random forests Support vector machines Deep learning
	Unsupervised		
	Clustering*	Clustering methods: K- and X-means, hierarchical Principal components analysis Deep learning * Since unsupervised methods do not describe a relation between a dependent and interdependent variable, they cannot be labelled linear or non-linear.	

Figure 1.1: Machine learning methods in financial services. Image taken from [4].

In Figure 1.1 it is possible to see that the combined approach of supervised and unsupervised methods is a valid approach for our type of research.
In the analysis conducted by the Italian Department of the Treasury, Ministry of the Economy and of Finance [5], a series of comparisons are made to attest the value of attributing a credit risk score such as the one we are conducting in this analysis. In figure 1.2 we can see how much the demand for credit and therefore its evaluation is of essential and how the economic and financial crisis of 2008 had

a series of repercussions on both the real economy and the credit market.
The risk of insolvency has grown and the risk analysis models of the institutes

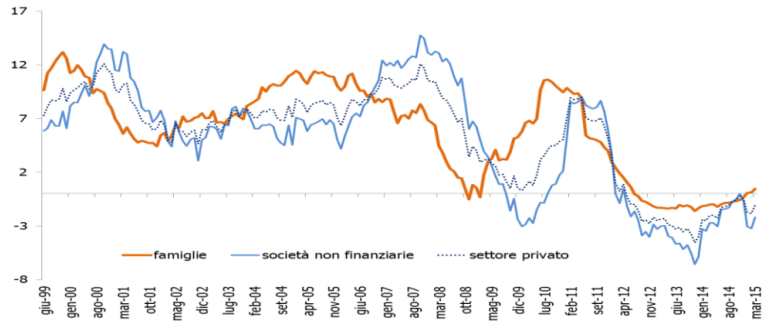

Figure 1.2: Trends of nominal loans to households (in orange), non financial corporations (in blue), and the private sector (dashed) between 1999 and 2015. y-axis: average annual change in nominal loans, x-axis: time. Image taken from [5].

in charge of this evaluation have not proved to be sufficiently accurate. This has led banks to increasingly stringent and rigid constraints. The credit granted by Financial Institutions (FIs) is an important tool for the economic development of the global market. The idea that credit influences the economic cycles is a concept already analysed by Bernanke (1983, 2000) [6] in the context of the Great Depression and in the "financial accelerator" theory. The analyses conducted in [5] have shown that loans in the private sector are significant also in anticipation of inflation and price trends. With the help of a Vector Autoregressive (VAR) model, described in its general form in Fig. 1.3, it is possible to identify the variables that contribute most to the explanation of credit and how it is possible to identify the existing relationships.

$$Y_t = c + A_1 Y_{t-1} + A_2 Y_{t-2} + \ldots + A_p Y_{t-p} + \varepsilon_t$$

Figure 1.3: Vector Autoregressive (VAR) model. Image taken from [5].

1.1.2 Income risk score

Figure 1.4 shows a flow chart of the analysis conducted by S. K. Annappindi, 2014, [7] to assess the credit risk score, similarly to how we will do in our analysis.

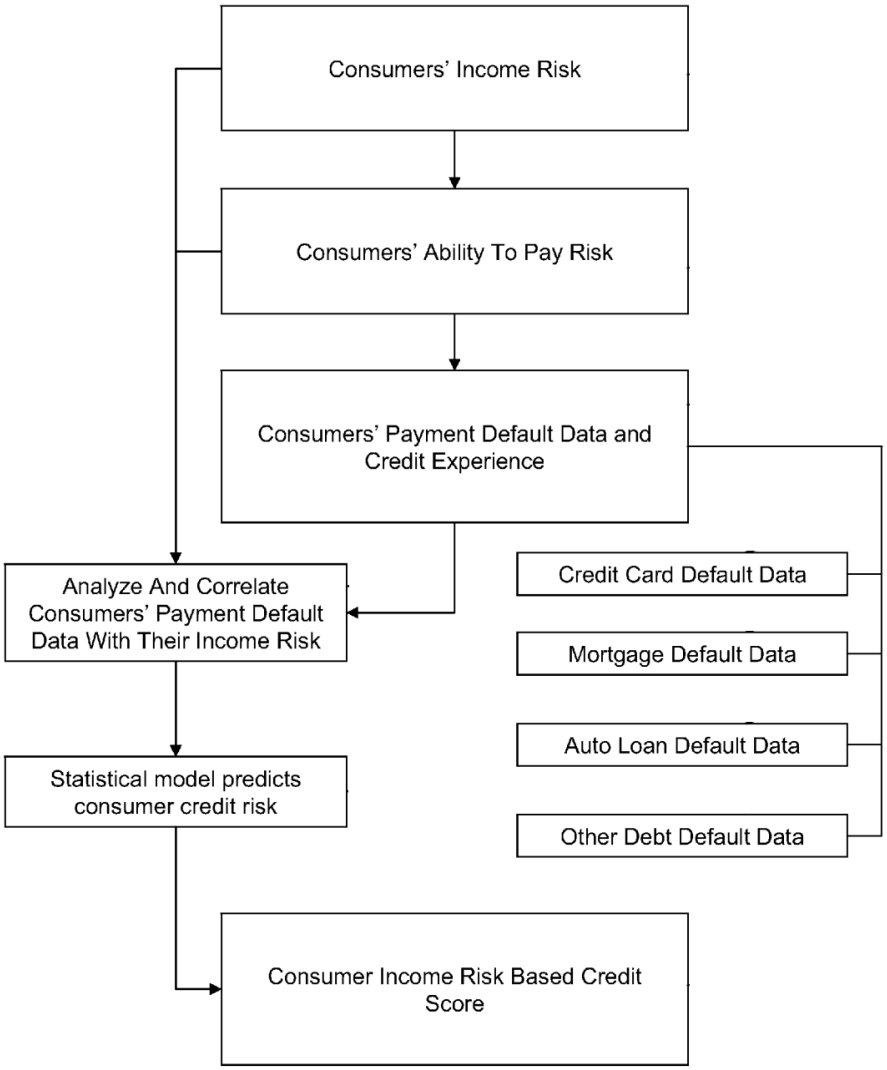

Figure 1.4: General flow chart to assess the credit risk score. Image taken from [7].

The assessment system proposed by [7] was finalized in a patent, in which systems and methods were described to assess the credit risk of consumers by analyzing their income risk and creditworthiness. The goal of the system is to estimate consumers' future ability to pay. This method allows us to make a prospective assessment of an individual's ability to repay a debt or ability to pay for products and services.

This study also compares the methods currently adopted by FIs and their limitations (Figure 1.5).

Existing Credit Scoring Models & Limitations

Credit Score	Description	Limitations
Credit Bureau Scores (FICO, Beacon, Vantage, etc)	• Based on consumers' credit bureau histories • No usable score if sufficient credit history is not present • Suggests future payment behavior will mirror past payment record	• Do not consider consumers' income disruption risk
Lenders' Internal Credit Scores and Custom Credit Scores	• Typically a variation of credit bureau scores • Based on consumers' credit bureau histories • Based on lenders' own portfolio credit performance data • Based on 3rd party portfolio data	• Do not consider consumers' income disruption risk
Alternative Credit Scores	• Based on consumers' non-credit payment histories including rent, phone bills, and utility payments • Not based on consumers' credit histories	• Do not consider consumers' income disruption risk

Figure 1.5: Currently adopted methods and limitations. Image taken from [7].

The Financial Risk Score is a new evaluation system that integrates the credit risk score and consumer credit score. A forecasting approach based on machine learning techniques allows to increase the accuracy of the forecasts on credit risk with a consequent increase in the quality of the credit offered, an increase in acquisitions and profitability in the consumer credit sector. The customer's ability to pay depends mainly on disposable income and spending habits.

The approaches adopted today by lenders are based on traditional credit scores. Traditional or Bureau credit scores are proving inadequate and non-descriptive for individuals with no credit history. In particular, they are unable to predict future spending capacities as they are based exclusively on the credit history and on the payment capacity, generally self-declaring, of consumers. The main problems are:

1. The scores are not reactive. They change only after the values have been

recorded, therefore only following a continuous and prolonged insolvency.

2. Income is often not considered, or is inaccurate. It is rarely possible to trace where the capital is invested.

3. It is not possible to assign a credit score to those without credit history.

1.1.3 FICO score

Financial risk represents an overall financial exposure score. Credit Score is a 3-digit score that can decide the fate of a credit application and in particular will determine interest rates. Furthermore, the rent or other requests could also be strictly connected to the analysis of the credit score. One of the most used is the FICO score: or Fair Isaac COrporation, credit scores are a methos of quantifying and evaluating an individual's creditworthiness. The score ranges from 300 to 850, with some lenders considering a score below 620 as subprime [8]. One of the main problems is the uniqueness of the FICO score; in fact, the three main credit agencies (Experian, Equifax and TransUnion) collect and report credit information independently. It is therefore possible to have a different FICO score for each office based on the data in their possession. The parameters (and their weights) used to calculate the FICO Scores [9] are shown in Figure 1.6.

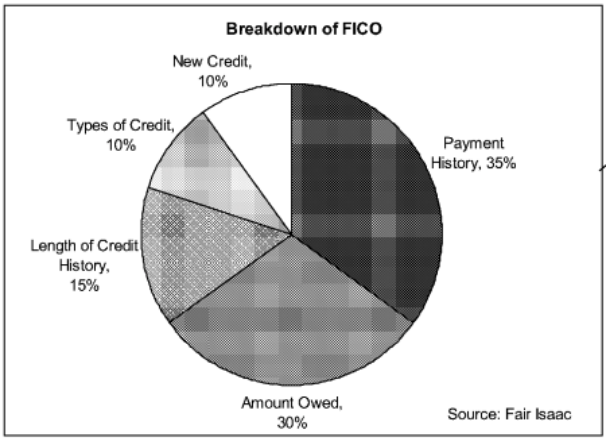

Figure 1.6: Parameters (and their weights) used to calculate the FICO Scores, taken from [7].

The first versions of FICO had a variable to measure the number of consumer credit cards. In the beginning (in 1992) having many credit cards was considered risky, while having few credit cards was considered a good behaviour. In a short time (in 1998) consumers with many credit cards were considered less suspicious

than the ones with only one credit card. This example shows how much there is a need to periodically retrain the scores in order to take into account changes in risk models as consumers behave differently. Credit requests have always been essential to assess the credit score. Although only a part of the final score, they have always been the focus of consumer attention. A first improvement to the FICO score was in introducing a buffer and a time window so that the latest credit requests would not affect the overall score. The buffer excludes the last requests and the time window aggregates the requests. The analysis showed that a larger window allowed an improvement even if the improvement was negligible for the cases of 60 or 90 days. For simplicity, a single window of 45 days was chosen (much more predictive than the initial 15 days), without differentiating between the cases among consumers (in fact, anyone with a bad credit history is reasonable to think that it takes longer to apply for a new credit).
Since the 2000s bank accounts have become increasingly complex. For example, flexible accounts have been created that have no limits except that of repaying the entire amount to the next cycle (generally the following month). These cases, together with others, have led to a more complex model called FICO 8.

FICO ranges from 300 to 900; The sector specific FICOs have a range from 250 to 900. They are divided as:

1. 800+ -> Exceptional
2. 740:799 -> Very Good
3. 670:739 -> Good
4. 580:669 -> Fair
5. 579 and lower -> Poor

It is evident that the credit score is a measure that allows you to evaluate the ability of the past; but what it would really matter to know is the ability of the future, therefore a forecast of ability.
A particularly interesting model was proposed in 2006 by Equifax, Experian and TransUnion, called VantageScores [10]. This score model, used in the context of unsecured loans such as credit cards, brings different advantages respect to FICO. In fact, the resolution of the credit is considered due both to the "ability to pay" and to the "willingness to pay". The ability to pay off a loan is proportional to the level of income while the willingness to pay is assessed on the basis of past payment behavior.

1.1.4 Machine learning for fraud assessment

Some important papers such as [11], [12], [13] and [14] illustrate how promising results have been gathered in the context of machine learning and neural networks in financial and non-financial fraud research.

In [11] a wide and varied comparative analysis is performed on the most used algorithms applied in the context of credit risk core assessment. Figure 1.7, shows some of the best results to date recoverable in academic and research.

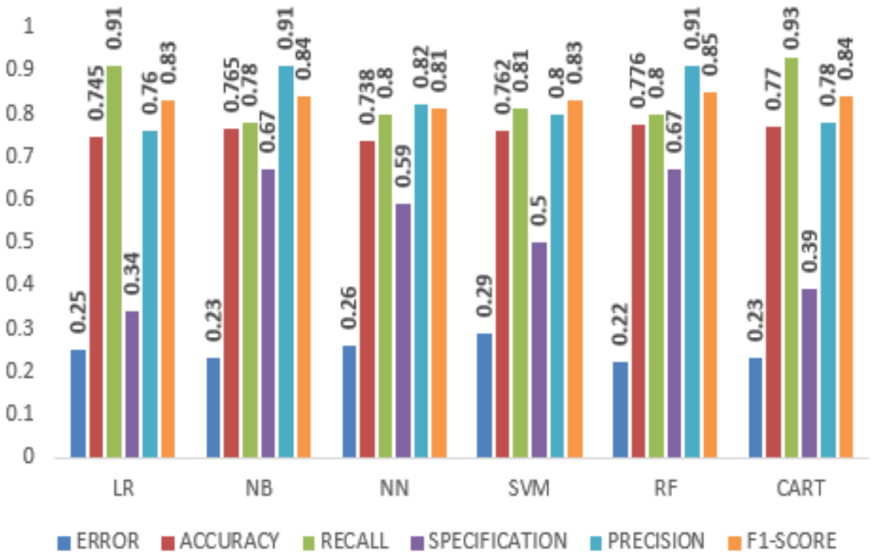

Figure 1.7: Bar diagram reporting the performances of different machine learning models to predict the credit risk core (Logistic Regression (RL), Naïve Bayes (NB), Neural Network (NN), Support Vector Network (SVM), Random Forest (RF) e Classification And Regression Tree (CART)). Image taken from [11].

These results allow us to easily understand that there is no globally accepted method that has been shown to be better than others except on specific datasets that are not particularly significant at the level of structured research in the field. Hence the importance of further analysis in this area by comparing different machine learning techniques to assess credit risk. In fact, global markets are full of risks and many attempts have been made to find quick and efficient ways to predict the future, so even an empirical or cross-sectional study like the one proposed in this thesis can improve the current credit and credit risk assessment scores that are of great benefit to the banking industry.

In [12], a set of machine learning models was tested in the context of credit deciding (i.e. the process of deciding whether to grant a credit card or a loan). The qualitative results of the research are shown in the table in figure 1.8, where we can

see that there is no particular improvement over the standard methods analysed in the previous article.

This work has two main limitations:

- The comparisons are made using data provided by two leading credit markets in Europe. Future works should conduct similar analysis in other geographic area.
- The study was done with three years of historical data, but since the market is changing rapidly we should continue to update it with continuously updated research. It is therefore recommended to repeat these experiments after a few years to better understand the progress of risk management digitization.

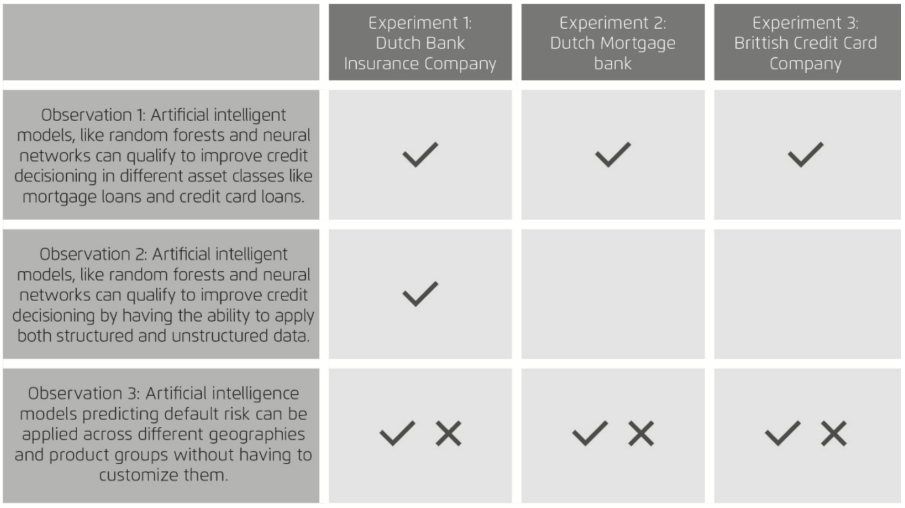

Figure 1.8: Qualitative Results of machine learning models for credit deciding. Image taken from [12].

As described in [13], current research in the area of risk in banking is much broader than what we are going to analyse in the course of this thesis. In fact, banks face many different types of risk: interest rate risk, market risk, credit risk, off-balance sheet risk, technological and operational risk, exchange rate risk, country or sovereign risk, liquidity risk and insolvency risk. The article goes on to detail the various types of risk defined in the annual report of 10 leading banks, shown in the chart in figure 1.9 that illustrates the taxonomy of the various types of risk.

In addition, the same article [13] addresses and describes the various methodologies or tools implemented to manage these risks (Figure 1.10).

The tables reported in the Appendix (A.1, A.2, A.3, A.4, A.5, A.6) provide a list of the articles that were reviewed by [13], classifying them by risk type,

including the risk management method/instrument and the algorithms. These researches were taken as examples for the study of models and methods that are implementable to our problem.

1.2 Partnership

In this dissertation, credit scoring models are proposed, using real payment data retrieved from a company in a PSD2 and open banking context. The PSD2 regulation is a European regulation that aims to make the management of money and payments safer and more convenient.

The research was conducted in collaboration with a company, which we will call for confidentiality SC (Smart Company). SC offers to the customer a service to monitor the activities of its bank accounts. Needless to say, the service collects the transaction data only from the bank accounts chosen by the user.

Transaction data are for example bank account transfers, electronic payments and/or credit/debit card payments. The cash flow statement can be calculated as the amount of collections or inflows minus cash payments or outflows over a specified period of time. This financial statement can be used to measure a user's financial strength or leverage. A positive flow, i.e. more revenue than expenditure, indicates favorable financial health. SC provided a data sets with transactions received from the banks that the user has chosen to connect. This allows to identify the main account and the movements between accounts linked to the same user (categorized as Transfers). The data of the single transaction include: the date of the transaction, the amount and some labels. These labels are provided by the categorization of the bank to which the account refers, so they may vary from bank to bank. These may include:

1. Account (75%),
2. Credit Card (8%),
3. Checking (7%),
4. Card (4%),
5. Savings (3%),
6. Debit Card (0.5%),
7. Loan (0.1%),
8. Investment (0.1%),
9. Mortage (<0.1%),

10. Credit (<0.1%).

SC also has its own system for the categorization of expenses and the possibility to correct this categorization through the user's suggestion. These are:

1. Food and Drink (20%),
2. Transfers (15,5%),
3. Goods (12%),
4. Fees (9%),
5. Transport (8%),
6. Earnings (7%),
7. Home and Family (6%),
8. Other (7%),
9. Leisure (5%),
10. Services (5%),
11. Wellness (3%).

Categories can allow data segmentation and provide useful trends over time.

In the end, it is then possible to imagine the analysis on a set of users as in the diagram in figure 1.11, each user can be considered fraudulent or not, depending on the associated accounts. Each bank account has a list of associated transactions described by the fields described in section 2.1.2.

1.3 Methods overview

The challenge is to create a new credit scoring system, called in this thesis Financial Score. This system differs from the logic adopted so far and is based on the actual financial behavior of users: constantly evolving and subject to sudden changes, so that it is as fair as possible and is not distorted by individual events but which evolves contextually with financial history. The challenge translates into being able to find a numerical estimate of the probability of solvency based solely on the individual's spending and earning behavior. The goal of the Financial Score is to be able to adapt the credit score to changes in consumer behavior by comparing with other users; with particular reference to the users most similar to him. Through

this process, the predictive scores developed will be increasingly sophisticated in recognizing consumers who manage their credit responsibly.

The SC company does not have access to the solvency history of users, information which can be requested only under the express intention of the user towards the competent authorities. From a technical point of view we are faced with a regression problem: given a generic set of characteristics and indicators in input, a Machine Learning algorithm must return a numerical value in output. The greatest difficulty lies in the training of the algorithm which cannot be supervised, that is, it is not possible to rely on already calculated scores associated with the examples contained in the dataset. That is, the inputs available do not have a label that contains the numerical value that the algorithm has the task of learning to predict. It is therefore an unsupervised regression case.

1.3.1 Supervised fraud analysis based on user transactions

In an initial supervised learning phase, this algorithm will use a fraud label extracted from internal research which has led to the classification of some users as fraudulent. The consumer population for which individual data has been provided can first be divided into two "fraudulent" / "non-fraudulent" categories. The distribution of cases is as follows:

1. 95% of the values are NaN (out of total transactions),
2. 5% of the values are "fraudulent" or "non-fraudulent" (out of the total transactions),
3. 0.8% of the values are "fraudulent" (out of total transactions),
4. 4.2% of the values are "non-fraudulent" (out of total transactions),
5. the "fraudulent" values compared to the "non-fraudulent" values are 18%.

Given the strong imbalance of data in this division, it is advisable to use special "imbalanced learning" libraries capable of increasing minority cases (in this case fraudulent) with mathematical techniques to improve learning. The nature of the fraud phenomenon is complex and the evolution over time can vary a lot: new frauds or more effective systems may arise to avoid being investigated. Due to the criticality in the automatic recognition of fraud, we choose to use this information in this context as indicative of incorrect behavior. In particular, the final Financial Score will be penalized in the event that a user has a time window categorized as fraudulent while he will have an advantage in the non-fraudulent case. This first supervised categorization allows us to assign a starting point to our financial score. Using a supervised method as a starting point allows you to keep the system

up to date by sector experts who can indicate some users or some transactions as "fraudulent" intended as a behavior from which to move away to increase your Financial Score. The Financial Score in fact wants to be representative of users' financial virtuosity.

1.3.2 Time series clustering

Due to the intrinsic difficulty of the problem, it is considered necessary to evaluate also a clustering phase to guarantee a more effective description of the distribution of wealth among users and consequently their spending behavior in the current society.
Clustering, in addition to being based on the statistics generated by the analysis of transactions (shown in table 2.1), uses the metric called "Dynamic Time Warping" and similarity "LB Keogh". At the end of this clustering phase, we obtained groups of users with similar transactional behaviour. In further studies, domain experts should classify certain behaviors and / or users as virtuous or fraudolent to confirm the applied clustering methods.

1.3.3 Time series forecasting

In addition, individual analysis were conducted for each user (providing statistics: seasonality, stationarity, trend and residual). The individual analysis of the statistics foresees a series of predictions on the trend of finances in the future on the basis of a historical learning. The models used are those that have had the best results in statistics and econometrics: based on the autoregressive integrated moving average (ARIMA Autoregressive integrated moving average) [15], and which have found the most success on the market (e.g. Prophet - Facebook) [16].

Individual analysis and critical aspects

The individual analysis is inspired by the SOW (Share of Wallet) invention, which refers in general to the processing of user financial data to define the credit score and a customer financial profile. In other worlds, SOW analyze the consumer behavior on the basis of his/her spending capacity [17], [18]. According to SOW's analysis, consumers will tend to spend more in proportion to the growth in their purchasing power, so a consumption model that can accurately estimate purchasing power is of fundamental interest to many financial institutions and other consumer services companies.
Analyzing the possibility of monitoring the credit balance from the accounts of its customers we note how often it is difficult to confirm that the balance is not affected by transfers from other accounts in different banks. These transfers do

not represent an increase in spending capacity and therefore this model is not sufficient for an exhaustive analysis. In addition, for each bank account, it would be necessary to differentiate between savings and investment. However, users are unlikely to have a single account or to share information protected by privacy. Not all the accounts of a user are linked to SC. In general, we can imagine a situation similar to the one depicted in Figure 1.12 where the user in question has linked three different bank accounts, each with a linked credit or debit card, possibly used for different purposes, e.g. money exchange between individuals, online payments, recurring payments, investments, purchase of real estate or luxury goods. These issues are also reflected in the incompleteness of the information held by the credit bureaus. Given the difficulty in identifying budget transfers, methods are needed to model consumer spending behavior. Consumer behavior can be modeled by first dividing it into categories of users (which can be based on budget levels, demographic profiles, income levels; these categories can also be monitored in a previous period of time).

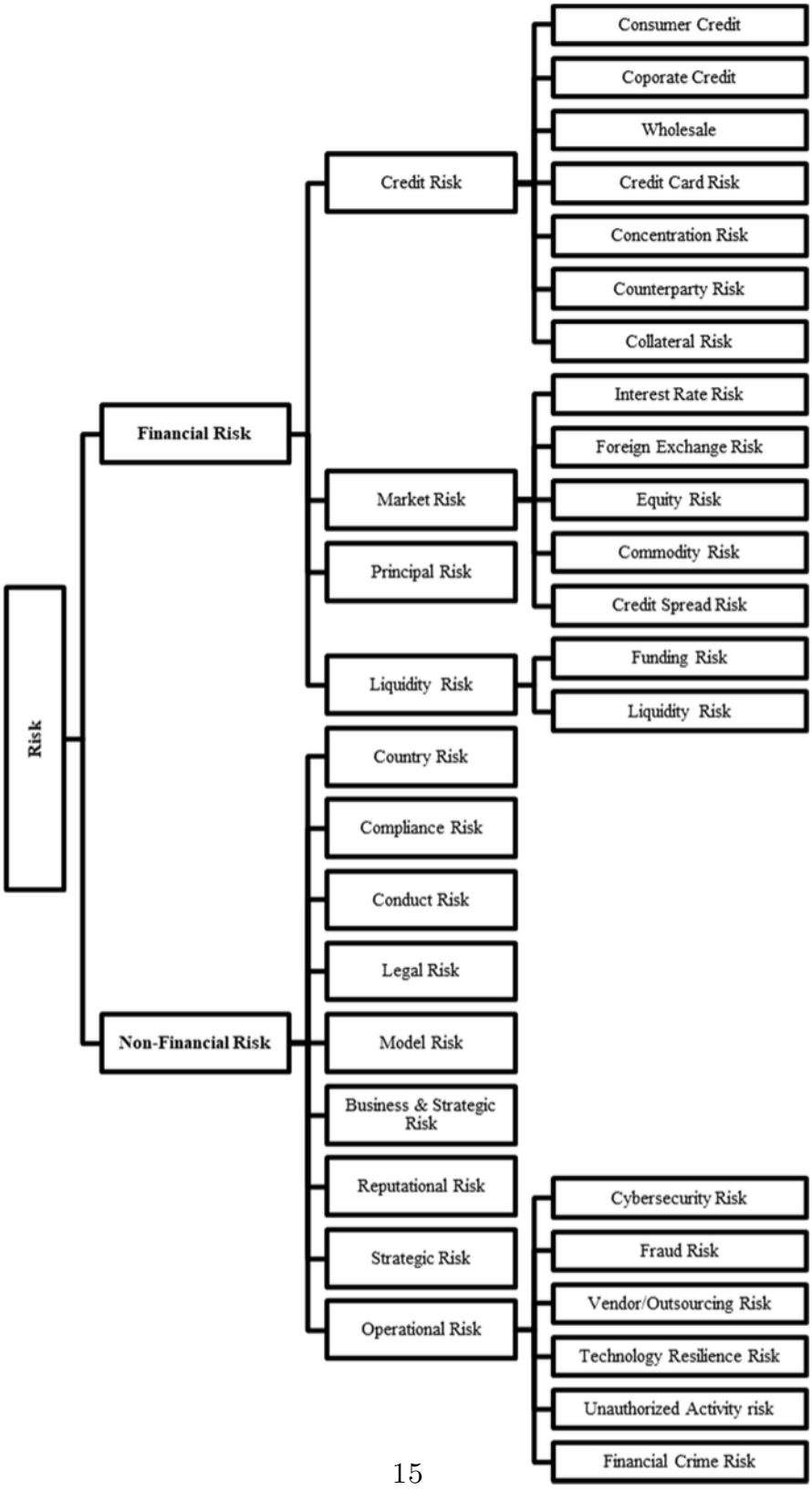

Figure 1.9: Taxonomy of the different risks incurred by banks, taken from [13].

	Market Risk	Credit Risk	Liquidity Risk	Non-Financial Risk (Operational Risk)
Risk Management Tools				
Risk Limits	√	√	√	
Credit Risk limits		√		
Value at Risk	√			
Earnings at Risk	√			
Expected Shortfall	√			
Economic Value Stress Testing	√			
Economic Capital	√	√	√	√
Risk Sensitivities	√			
Risk Assessment (RCSA)				√
Operational Risk Losses				√
Loss Distribution Approach				√
Scenario Analysis	√	√	√	√
Tail Risk Capture	√	√	√	√
Stress Testing	√	√	√	√
Scoring Models		√		
Rating Models		√		
Exposure **- Probability of Default** **- Loss Given Default** **- Exposure at Default**		√		
Back Testing	√	√	√	
Risk Management Framework Components				
Risk Appetite	√	√	√	√
Risk Identification	√	√	√	√
Risk Assessment	√	√	√	√
Risk Measurement	√	√	√	√
Risk Testing	√	√	√	√
Risk Monitoring	√	√	√	√
Risk reporting	√	√	√	√
Risk Oversight	√	√	√	√
Capital Management (calculation and allocation) **- CCAR** **- ICAAP**	√	√	√	√

Figure 1.10: Risk Management Methods and Tools, taken from [13]

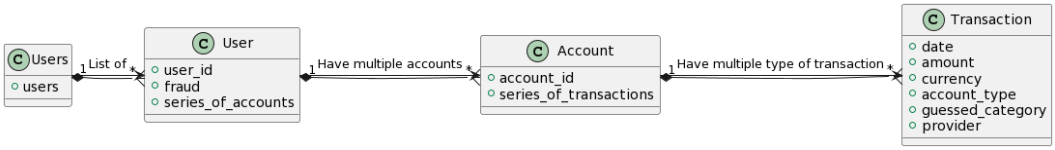

Figure 1.11: Users overview, from left: all users, each user has multiple bank accounts with multiple associated transactions

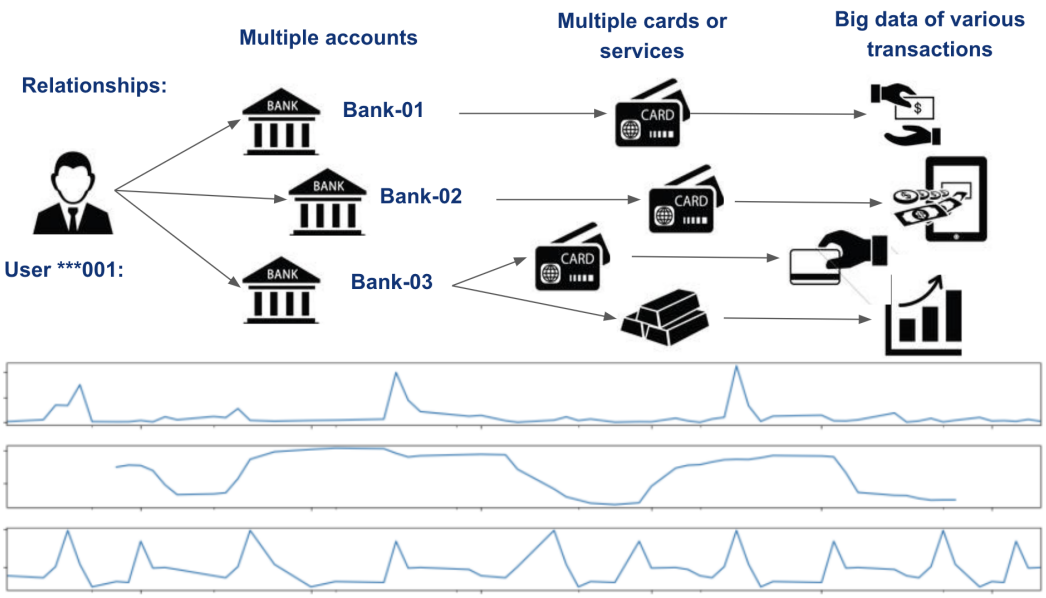

Figure 1.12: Example of User Account Situation

Chapter 2

Data Representation

2.1 Data description

Through the regulations introduced by the European PSD2 directive, which regulates payment services and payment service providers within the European Union, it is possible, among the various innovations introduced, to access information on online payment accounts on the basis of which it is possible to obtain aggregate information on one or more online accounts held even by different institutions (obviously only with the consent of the account holder).
One of the main challenges in adopting the European PSD2 regulation by banks has been to adapt and integrate their systems to meet generic standards that are more easily understood by automated systems capable of processing them. In fact, the regulation does not go into specifics about the data representation of transactions or banking operations that can be accessed. Generally, this aspect of the representation of transactions is modeled by the companies that offer this service of reading and aggregating information about the various accounts.

2.1.1 Data overview

It is possible to imagine the dataset as a set of transactions (Transactions box, the first from the left, in the UML diagram of figure 2.1), these are associated through a unique id to an anonymized user (User box, the second from the left, in the UML diagram of figure 2.1) who is therefore considered fraudulent or non-fraudulent (although this information is not available for most users). Each user is also associated with multiple accounts (Account box, the third from the left, in the UML diagram in figure 2.1) and thus it is possible to imagine the single Transaction as a series of transactions associated with the syncular user (Transaction box, last on the right, in the UML diagram in figure 2.1).

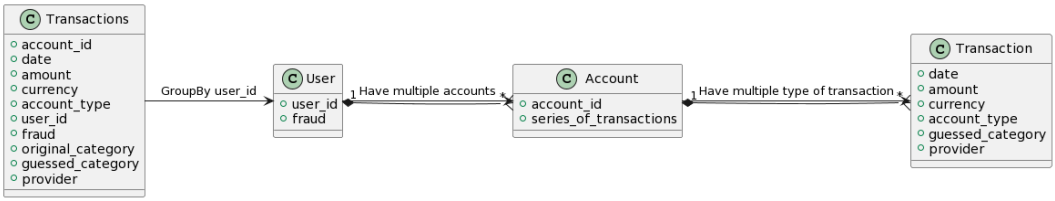

Figure 2.1: Transactions overview, from left: raw transactions as per the source dataset that are represented as linked to a user who has multiple accounts with multiple associated transactions

2.1.2 Fields

The dataset consists in a set of variables for each transaction. Each transaction is linked to an account. A single user can have multiple of accounts of different type. Figure 2.2 shows an overview of the amount of information available in the dataset under analysis. The description of the single variables is reported in following chapters.

Number of variables	10
Number of observations	3817722
Missing cells	3634608 (9.5%)
Duplicate rows	128540 (3.4%)
Total size in memory	291.3 MiB
Average record size in memory	80.0 B

Figure 2.2: Overview of the Dataset

User Id

Identifier of the user who made the transaction.

user_id
Numeric

Distinct count	31313	**Mean**	2150095182
Unique (%)	0.8%	**Minimum**	211993
Missing (%)	0.0%	**Maximum**	4294673894
Missing (n)	0	**Zeros (%)**	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

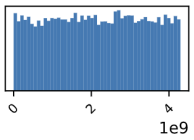

Figure 2.3: User Id

As it is possible to see from figure 2.3 the analysis is based on a total of beyond 31 thousands users.

Account Id

Uniquely identifies a user's portfolio.

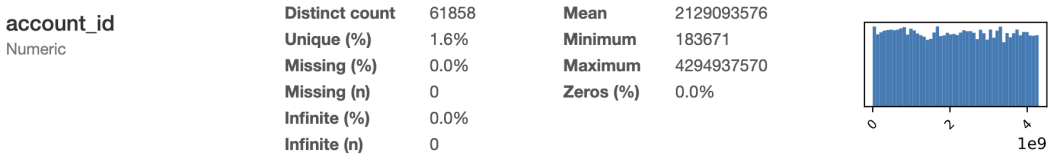

Figure 2.4: Account Id

From Figure 2.4 we see that in total we have almost 62 thousand accounts available.

Account Type

Describes the type of account.

Figure 2.5: Account Type

From figure 2.5 we notice that most of the transactions are toward the account type "account" (77.4% of the total transactions). In fact, "account" refers to a generic money transfer toward a bank account. "credit_card" (e.g. a payment with a credit card) and "checking" are the other values that exceed 5%, 8.1% and 7.4% respectively. The other values cover less significant percentages: "card" 3.9%, "savings" 2.6%, "debit card" 0.5%, "loan" 0.1%, "investment" 0.1%, "mortgage" <0.1%, "credit" <0.1%.

Amount

Positive real number describing the amount of the transaction.

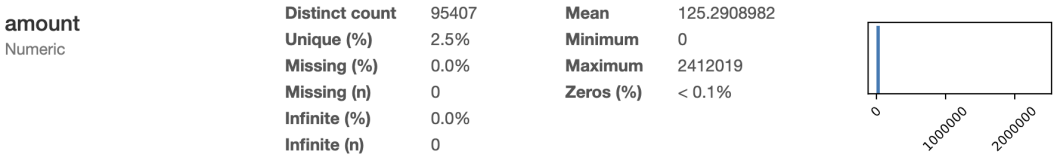

Figure 2.6: Amount

From figure 2.6 we note that, as expected, many transactions have the same amount and only a small subset (< 0.1% of the total transactions) have a zero amount (these transactions are assimilated to validity checks on the account or card). Validity check transactions can also assume values close to zero (< 0.04 euro). 5467 transactions with amount 0.01, 2377 transactions with amount 0.02, 1741 transactions with amount 0.03, 1535 transactions with amount 0.04.

The maximum amount among all the transactions (2412019 euro) is unlikely, probably due to errors or application imperfections (of the bank itself or of the reading system), in fact, we find 1 transaction with amount 2412019, 1 transaction with amount 451666.67, 1 transaction with amount 445003, 1 transaction with amount 444000, 2 transactions with 400000, etc...

Currency

In the context of the analysis of European transactions we have two types EUR (euro) and GBP (pound sterling).

Figure 2.7: Currency

From the figure 2.7 can see how the transactions are rather balanced with respect to the currency in fact we have 56.4% in EUR and the remaining 43.6% in GBP.

Date

Day on which the transaction took place or was recorded.

Figure 2.8: Date

The most critical aspect of the date variable is that it has no information about the time within the date. As per the figure 2.8, the data provided for analysis is from the first 3 months of 2019. Another problem of uncertainty is the representation of the date by the bank, some may mean the execution date and other banks may mean the accounting date (which for certain transactions could vary up to 2/3 business days).

Fraud

Binary label that indicates that the user who made the transaction is involved in some kind of fraud or had all the prerequisites to commit one.

Figure 2.9: Fraud

As might be expected, we are faced with a highly unbalanced problem 2.9 where fraudsters are a tiny fraction of the total. In particular, we note that True fraud is only 0.8%. False frauds are users that for various reasons have been analyzed manually and it has been decided to exclude them and mark them as not-fraudulent. In the remaining cases, 95.1% of the transactions, since it is not possible to assign a label without additional legal information not in our possession, it is correct to assume that they are not-fraudulent users, but there is no guarantee that there are no fraudulent users among them who have not yet been detected.

Guessed category

Categorization of the transaction suggested by the user.

Figure 2.10: Guessed Category

Original category

Automatic categorization of the transaction inferred from the transaction description text field (not provided for privacy reasons).

Figure 2.11: Original Category

Comparing the two images on the transaction category, 2.10 and 2.11, we immediately notice that several tens of thousands of transactions are categorized

differently by the user (guessed category) than by the automatic system (original category). This suggests that this information is not reliable and usable for our analysis purposes. (Figure to show the distribution between fraudulent and non-fraudulent that have changed categories).

Provider

Identifier of the bank where the account is hosted.

Figure 2.12: Provider

From 2.12 we have highlighted the main bank providers that collect more transactions: bank-10 9.%, bank-05 9.1% and bank-37 8.4%.

2.2 Data preprocessing

Introduction

Data analysis begins and has its foundation in the initial phase of data preparation and cleaning. This is generally the most delicate phase of a machine learning study as all subsequent phases are strongly affected by this preparatory phase. Since the case study is a primacy of its kind, in an experimental context of analysis, with combined supervised and unsupervised techniques, it is very difficult to simply exclude less significant data. In fact, the data under analysis are obtained in an Open Data context, that is, only information transferred by consumers. Consent to read this data is explicitly provided by the user, it can be revoked at any time by the user, by the bank or simply expire after 90 days. This information is very difficult to obtain and in many cases downright impossible.
Furthermore, the economic system and the circulation of cash money are still today some of the most difficult information to validate and trace. The excessive use of cash payments, when detected, tend to lower the Financial Score, as it is considered a suspicious behaviour [19]. The incompleteness of information relating to consumer cash transactions is intrinsic in the system and cannot be resolved.
Moreover, there is the problem of duplicates and transfers: in addition to internal transactions between proprietary accounts and transactions between private individuals there are several services that introduce redundancy in data. While the transfer between accounts rather than to a merchant can be identified through the evaluation of the destination identifier, the exchange between accounts can generate

an alteration of transactions such as to alter individual analyzes. In general they introduce redundancy those services connected in reading to multiple accounts by adding particular transactions with the addition of virtual credit in the various wallets. The data are understandable in nature and simple to represent for their daily use to which we are accustomed. In the protection of privacy, all sensitive information is excluded, therefore the meaning of the Financial Scoring cannot be an exhaustive representation of the information available to credit institutions.

Data preprocessing

Data preprocessing can refer to the manipulation or elimination of data to improve performance and is one of the most important steps in data mining and machine learning. In fact, data collection methods, as in our case, are not sufficiently controlled by introducing out-of-range values or impossible data combinations, duplicate or missing data, etc. When a lot of information is irrelevant or redundant or you have particularly noisy and unreliable data, knowledge discovery during the training phase is more difficult. The data preparation and filtering phases can be resource-intensive and time-consuming to complete, but provide significant advantages in the subsequent steps. In the case of our dataset in particular, it is not possible to know if the value of the amount in question is particularly distant from the others due to a reading error or other. But it is reasonable to suppose that the bank has cancelled or blocked transactions with disproportionate amounts on the basis of the limits of the current account in question.

Transactions of particularly high amount can be associated with a wrong reading or with the purchase of real estate (or large business expenses), but in any case they are always subjected to other kinds of controls and checks by the competent authorities. From figure 2.13 we note that there are no fraudulent transactions greater than 16000. Instead, non frudolent transactions can assume values greater than 20000. It was therefore decided to remove all transaction greater than 20000 as they would be subjected to check anyways. This operation resulted in removing 1170 unlabeled transactions, 62 non-fraudulent transactions and 0 fraudulent transactions. In total, we exclude 1232 transactions for amounts that are too high compared to the transactions of fraudulent origin of our interest.

Analyzing the lower margin, it is possible to remove all transactions less than 1 euro, assuming that most of them are micro-transactions to verify the operability of the account, or small payments that are not similar to large transfers. In fact, banks do not normally allow transfers below this amount, and it is generally not possible to place orders online. This resulted in removing 151100 unlabeled, 6619 non-fraudulent and 960 fraudulent.

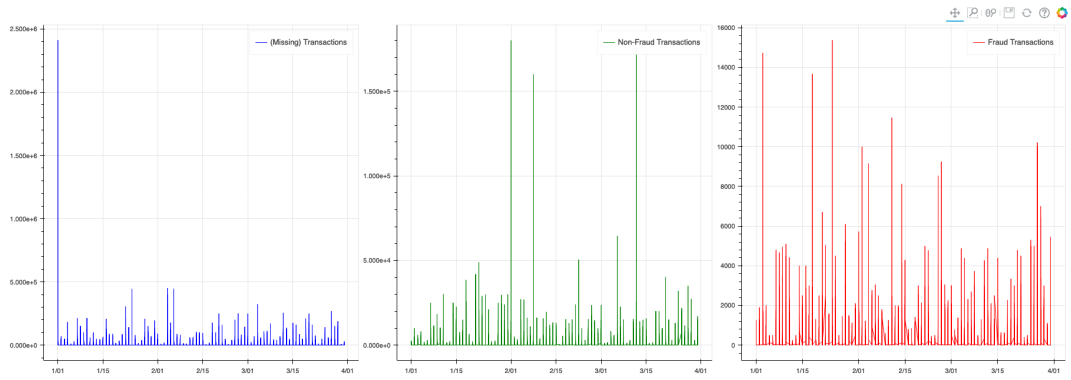

Figure 2.13: x-axis: maximum amount among all the transaction occurred in a day y-axis: time (days). From sx to dx: transactions missing the value of the "Fraud" field (blue), non-fraudolent transactions (green) and fraudolent (red).

Data deduplication

Data deduplication is a technique for eliminating duplicate copies of repeated data. The process of deduplication requires the comparison of "blocks" of data that in our case can be traced back to individual operations or algebraic sums between them. As a first approach, an interactive analysis of the bank providers was conducted on various users, identifying "bank-05" as the source and cause of most of the duplicates. Subsequently, all users with a "bank-05" account and other related accounts were processed through batch processing scripts to perform the necessary corrective operations. In fact, "bank-05" proved to be the service for digital payments called "PayPal" that being an additional service to be linked to the user's bank account introduced the same transaction with inconsistency problems on the other labels. In fact, the same transaction could be recorded with different dates due to the delay in paying the amount advanced by PayPal, or could be paid in part or completely with the remaining credit, or could introduce discrepancies in categorization due to a different description of the transaction. Analyzing the service, it proved to be non-trivial to compare the transactions aimed at identifying the account or accounts associated with the PayPal account. In fact, before excluding transactions for the user who has a PayPal account, it is necessary to identify the linked account(s) and verify that there are actually duplicate transactions. In fact, in the extreme case in which the user has connected only the PayPal account and not the associated bank account, it is appropriate to keep these transactions because they are not duplicated by other information. In the classic case in which a user has connected only one account with the PayPal account and both transactions are available, the

PayPal transactions will be ignored and therefore deleted. In the intermediate case when a user uses the PayPal account from several bank accounts, it is advisable to verify that all accounts are present, otherwise it is preferable to delete only the PayPal transactions corresponding to accounts where it is possible to read all transactions.

2.2.1 Accounts statistics

Once have removed outlier and duplicated transactions, the users behaviour was analyzed through a statistic analysis to have an overview of each account behaviour. Specifically, for each account of each user, the transaction history was processed to perform the statistics described in Table 2.1:

Parameter	Description
count_nonzero	**Non-zero amount number**
amax	**Maximum transaction amount**
amin	**Minimum transaction amount**
mean	**Mean of the amount of transactions**
std	**Standard deviation of the amount of transactions**
median	**Median of the amount of transactions**
var	**Variance of the amount of transactions**
sum	**Sum of the amount of transactions**

Table 2.1: Description of statistics parameters

	User#1		*User#2*		
User ID	***993		***373		
account ID	***872	***038	***495	***183	***345
count_nonzero	5	52	4	443	234
amax	500.00	271.49	300.00	1848.06	350.00
amin	28.11	0.99	0.41	0.60	0.44
mean	236.274	48.216	150.103	63.065	21.135
std	203.984	57.704	147.056	203.360	41.914
median	251.500	28.080	150.000	14.970	2.395
var	41609.419	3329.800	21525.709	41335.546	1756.766
sum	1181.37	2507.24	600.41	27937.63	4945.51

Table 2.2: Statistics calculated for two example users

In general, there are more accounts associated to a single user. As example,

Table 2.2 shows the statistical parameters calculated for 2 users. In this case, user#1 has 2 linked accounts while user#2 has 3 accounts. The statistical parameters describe the spending behavior of the individual.

Often, a user is associated with multiple accounts but only one of them is meaningful (primary account). The account statistics can be useful to distinguish the primary account from the secondaries accounts, as, for example, the latter will tend to have a lower variance. Figure 2.14 shows an example of a primary (blue) and secondary (red) accounts of a user.

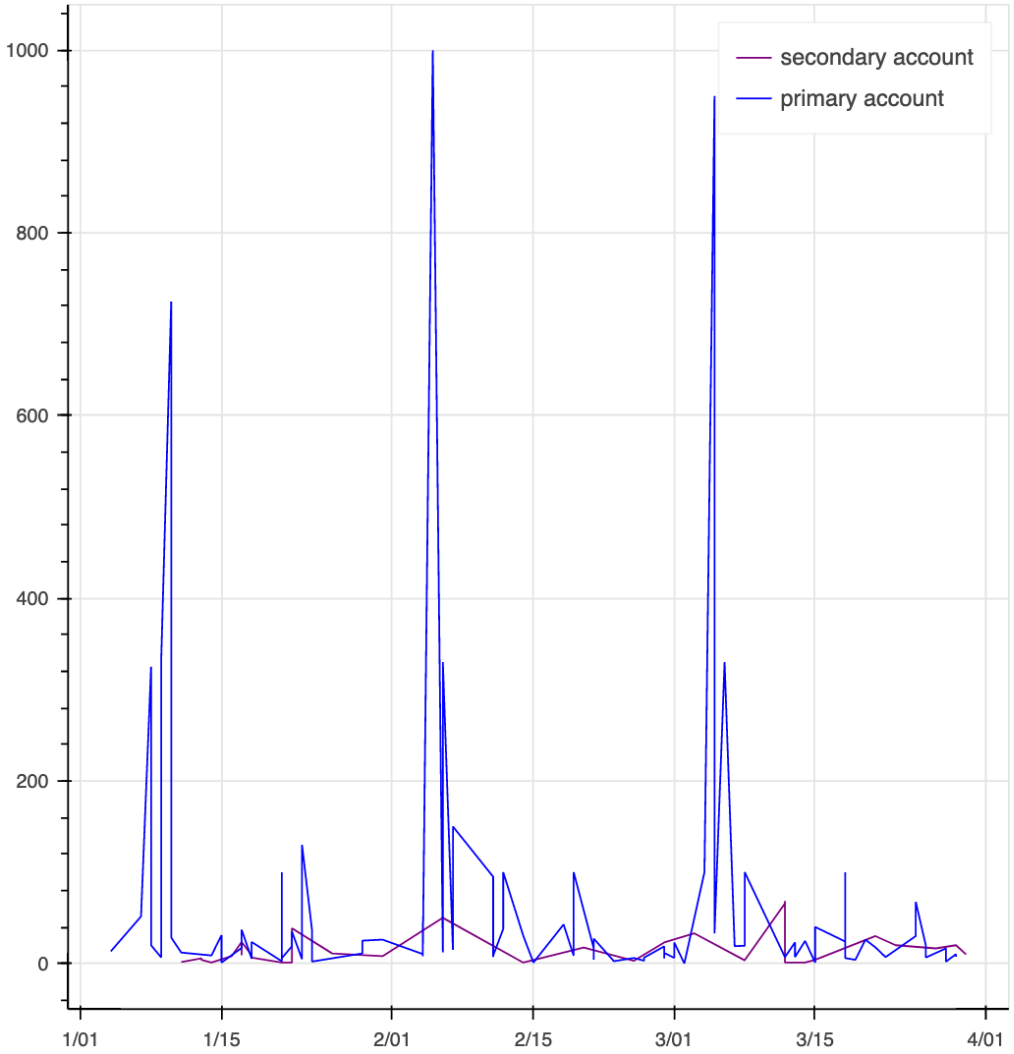

Figure 2.14: Example of a primary (blue) and secondary (red) accounts of a user.

2.3 Time series

Time series, generally speaking, in mathematics, is a series of data points indexed in temporal order. Thus a discrete time sequence of data points. It is most often plotted using a running graph (time line graph). Time series analysis includes methods for analyzing time series data in order to extract meaningful statistics and other interesting features of the data. Time series forecasting is the use of a model to predict future values based on previously observed values. It differs from regression analysis, which is often used to test relationships between one or more different time series. This is not time series analysis, which refers specifically to relationships between different points in time within a single series. The time series data has a natural temporal ordering of points. This distinguishes time series analysis from cross-sectional studies, where there is no natural temporal ordering of observations. A stochastic model for a time series will generally reflect the fact that observations that are close in time will be more closely correlated than those that are further apart. In addition, time series models will often make use of the natural unidirectional ordering of time so that values for a given period are expressed as arising in some way from past values rather than future values.The dataset under analysis contains for each transaction a label user and a label fraud.The fraud label was manually assigned by the company during an independent research. By aggregating all the transactions with the same label user, we create the user time series. A user is considered fraudulent because all his transactions were labelled as fraudulent, even though this could be not representative of the reality. In fact,in a real world scenario it is possible that a fraudulent user have also non-fraudulent transactions.

The representation that can be used for the analysis of our problem is a time series representation where each transaction is represented as its amount for a certain date (or time instant). Figure 2.15 shows the amount of the transactions over time (time series) of three different user. All the transactions belonging to the first user (in blue) are missing the value of the "Fraud" field, while all the transactions of the second (in green) are non-fraudolent and all the transaction of the third are fraudolent (in red). We can observe that the three time series are not easily distinguishable from each other.

Types of Time Series

In general, we can divide time series into two main types:

Univariate time series Univariate time series refer to a type of time series represented by a single value recorded in sequential order with equal or unequal time intervals. When you want to model a univariate time series, each metric is representative of changes in a single time-dependent variable.

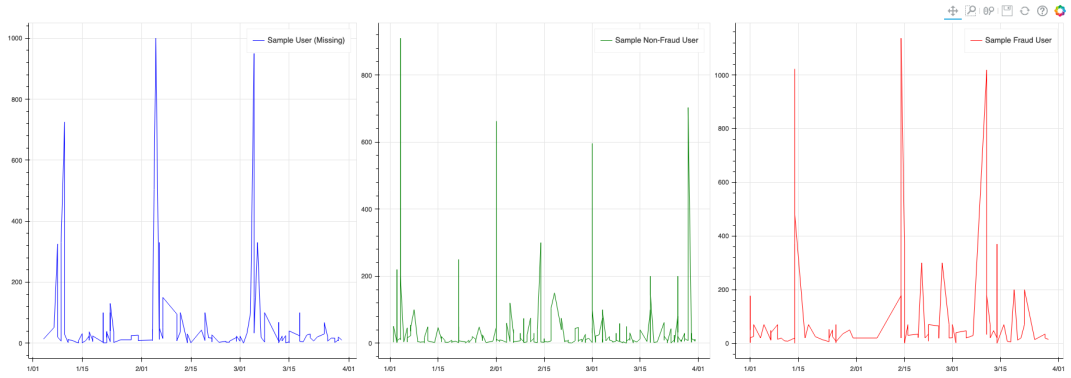

Figure 2.15: Example of time series of three users

Multivariate time series Multivariate time series, as their name implies, refer to a type of time series which can be represented by several time-dependent variables. In the case where there are exactly two time dependent variables then we can talk about "bivariate time series". In general in every multivariate time series its metrics have a certain dependence on the other variables. For example in our example image we can see a multivariate time series with n subplots of time series data that are used to predict individual's spending behavior.

2.3.1 Time series components

A time series have two main characteristics:

1. Measurability: that is, that a numerical value can be associated for each point.

2. Variability: that is, the metric changes over time.

Each piece of numerical data is associated with a time stamp and one or more labeled dimensions associated with the metric. It should be noted that the time intervals between each point can be regular intervals, or irregular intervals, as in our case in which each point (transaction) can occur at any instant of time more or less close to the previous and the next.

According to the additive model proposed by Harvey and Peters, 1990 [20], time series $(Z(t))$ can be decomposed in three components:

$$Z(t) = M(t) + S(t) + R(t)$$
$$t = 1, ..., T$$

Where:

1. M(t) is the trend, i.e. the overall direction of the data;

2. S(t) is the seasonality, i.e. the periodic component that indicates recurrent pattern over a period of time, e.g., every year, month, week, or day.

3. R(t) is the Residual, also known as "noise" or "volatility", which refers to random variations

Figure 2.16 shows a result of a time series decomposition.

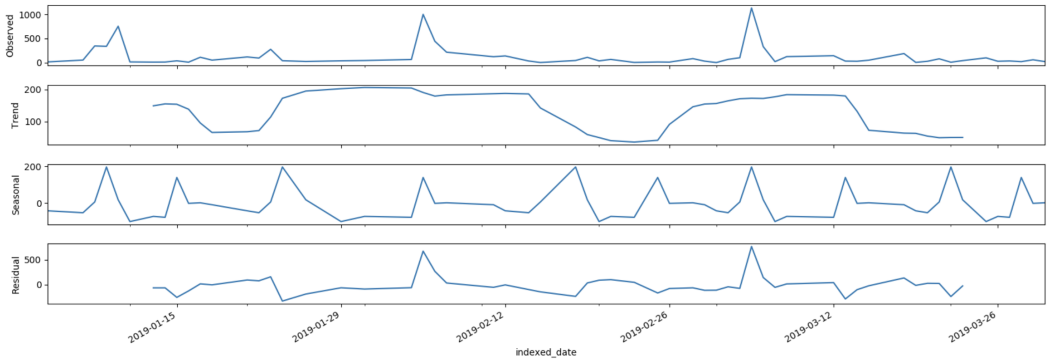

Figure 2.16: Example of a time series decomposition. From top: original time series, Trend component, Seasonal component and Residual component

Detection and forecasting of time series anomalies

One of the main applications of time series components analysis is time series forecasting. In particular, we can observe two main applications:

1. Time series anomaly detection: i.e., data mining techniques used to detect outliers in a dataset. We focus on this because among its main real-world applications are income and spending capacity monitoring and fraud detection.

2. Time series forecasting: i.e., the use of machine learning models to make predictions about future values based on previously observed data. In our case it is very interesting to make predictions on the trend of the individual's transactions in order to better assess the trend of his future score.

Let's look at the five key concepts for detecting and forecasting time series anomalies:

1. Seasonality: Finding the presence of variations that repeat or occur at regular intervals is common in real and financial data, and identifying these patterns helps improve our efforts to detect and predict anomalies.

2. Stationarity and non-stationarity: The common ideal assumption for time series techniques is that they are stationary, meaning that the statistical properties of mean, variance and autocorrelation are constant. Conversely, non-stationary time series refer to data whose statistical properties change over time. As we will see from the results that we will illustrate later in our case we will have several time series, some stationary, some stationary only for certain periods of time, some completely non-stationary.

3. Trends: It is important to record upward trends (which will improve the score) or downward trends (which will worsen the score) because it is a fundamental part of the prediction and detection of anomalies or fraud in the set of transactions.

4. Temporal pattern analysis: Time patterns refer to a signal segment that repeats in a time series and identifying them plays an important part in the analysis and categorization of time series.

5. Event impact analysis: In general we will look in the time series data if there are events that distort the data leading them to deviate towards a certain direction thus identifying their impact by adding value to anomaly detection and prediction.

2.3.2 Metrics for time series comparison

DTW - Dynamic time warping

Dynamic time warping, or DTW, is an algorithm that enables alignment between two time series [21], allowing a measure of distance between them. In particular it is useful to treat sequences whose components have characteristics that vary over time, and therefore where the only linear expansion or compression of the two sequences does not bring sufficient results. In general it allows us to find an optimal correspondence between two sequences, through non-linear distortion with respect to the independent variable (time). Generally, some restrictions are applied to the calculation of the correspondence: the monotonicity of the correspondences is guaranteed and the maximum limit of possible correspondences between contiguous elements of the sequence. For the sake of completeness, an implementation of the calculation of a distance measure based on DTW is reported, although during batch processing an equivalent version, implemented in optimized Python libraries, was used.

Listing 2.1: DTW Distance as implemented in [1]

```
1 int DTWDistance(int s[1..n], int t[1..m]) {
2   declare int DTW[0..n, 0..m]
```

```
3     declare int i, j, cost
4
5     for i := 1 to n
6       for j := 1 to m
7         DTW[i, j] := infinity
8
9     DTW[0, 0] := 0
10
11    for i := 1 to n
12      for j := 1 to m
13        cost:= d(s[i], t[j])
14        DTW[i, j] := cost + minimum(DTW[i-1, j  ], // insertion
15                                    DTW[i  , j-1], // deletion
16                                    DTW[i-1, j-1]) // match
17
18    return DTW[n, m]
19  }
```

Distance metrics will be used in the clustering phase to group users with similar behaviour and to analyze how a user behaviour changes among different period of time (e.g. month) or among different accounts.

The image 2.17 shows a comparison using the above algorithm that can be used to measure all the various metrics analyzed such as user comparison, for each account for each month. As stated in various literatures including [22] we can normalize the time series to select more meaningful information for our analysis. We will, however, retain the non-normalized measures as further verification of the process. The normalized measurements are depicted in the image of 2.18, which captures the same time series as image 2.17.

LB Keogh

The most satisfactory results were obtained through the use of DTW through LB Keogh similarity, which for our case under analysis showed better performance in terms of execution time for the same results.As described by one of the leading articles on the subject, cite, LB Keogh is a tool for the lower bound of various time series distance measures. It is one of the first non-trivial lower bounds for Dynamic Time Warping (DTW), and there are no faster techniques yet to index DTW ([23]).

We can describe its operation as an outer protective envelope built around the red time series (in the right section of the figure, blue in the left section), the Euclidean distance between the black time series (in the right section of the figure, orange in the left section) and the closest part of the protective envelope is a lower (tight) bound to DTW.

In the image 2.19 we show the comparison between two users using the LB

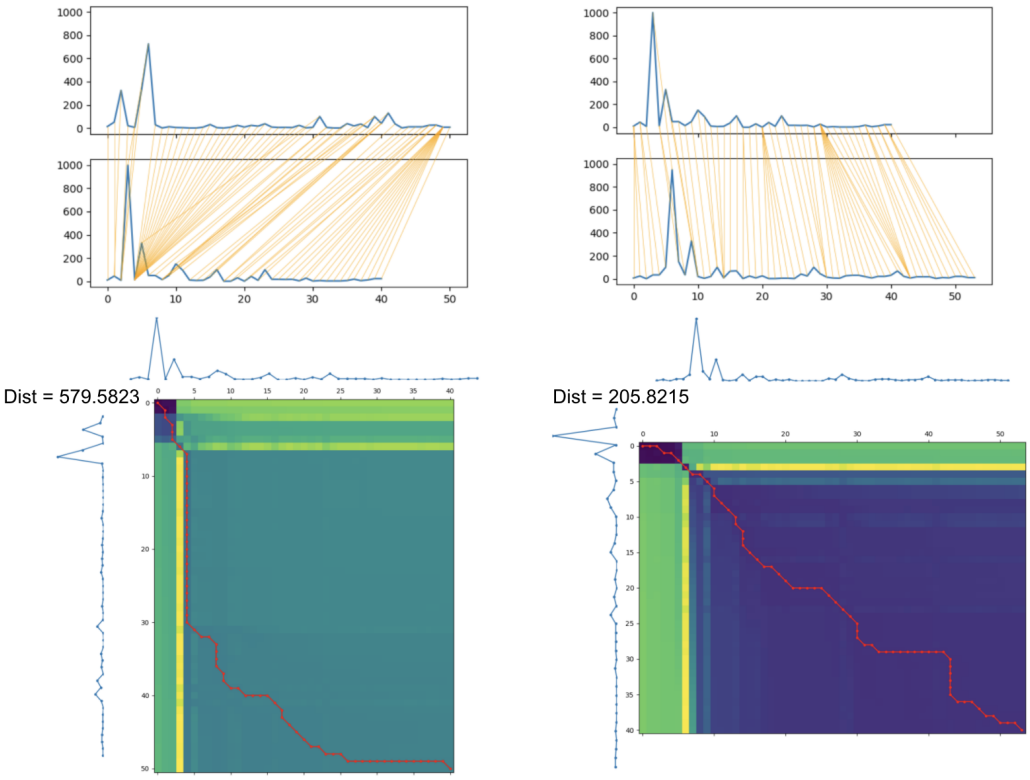

Figure 2.17: DTW measurement of monthly banking time series

Keogh similarity method. [24]

Matrix Profile

In this time series analysis we are interested in two aspects: anomalies and trends. One method of finding anomalies and trends within a time series is to perform a similarity join. That is, we compare fragments of the time series with themselves, calculating the distance between each pair of fragments. Although it is sufficient to implement nested loops, these can be particularly onerous in terms of time and resources. Taking advantage of Matrix Profile algorithms drastically reduces computational time [26].

The Matrix Profile is a relatively new, introduced in 2016, data structure for time series analysis developed by Eamonn Keogh at the University of California Riverside and Abdullah Mueen at the University of New Mexico [27]. Some of the advantages of using the Matrix Profile is that it is domain agnostic, fast, provides an exact solution (approximate when desired) and only requires a single parameter.

The matrix profile has two main components: a distance profile and a profile index. The distance profile is a vector of minimum Z-normalized Euclidean distances

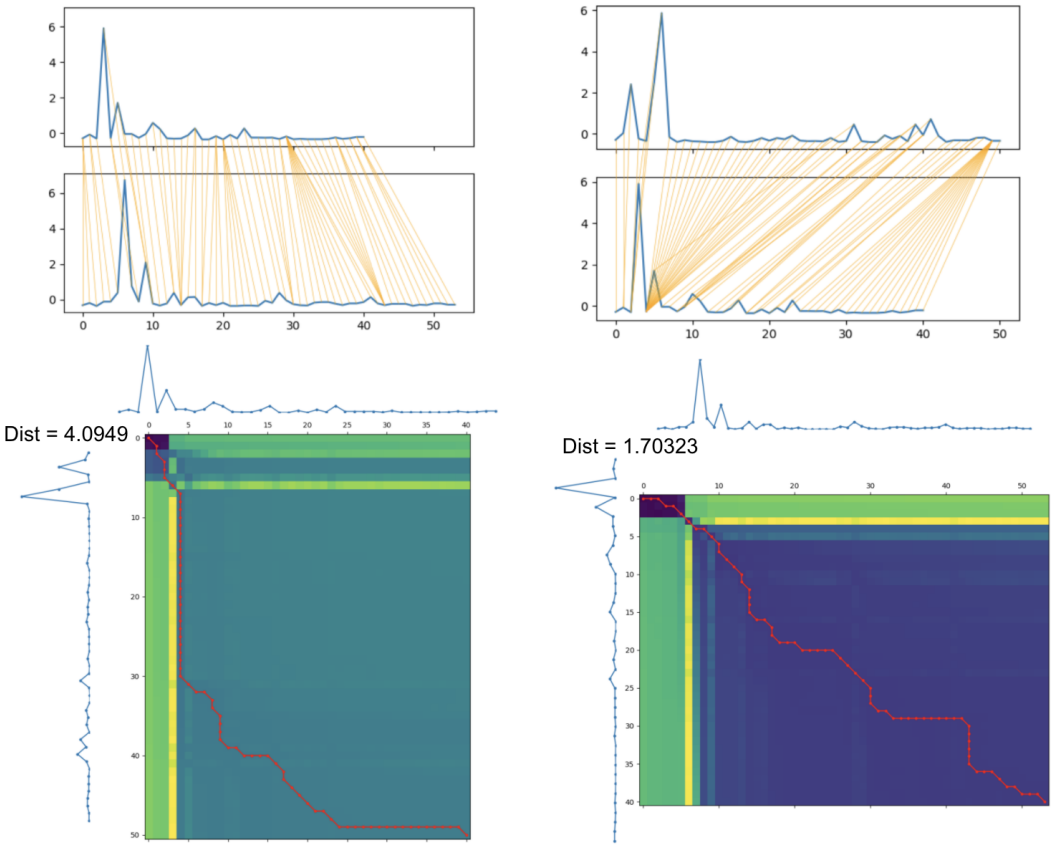

Figure 2.18: DTW measurement of monthly banking time series normalized

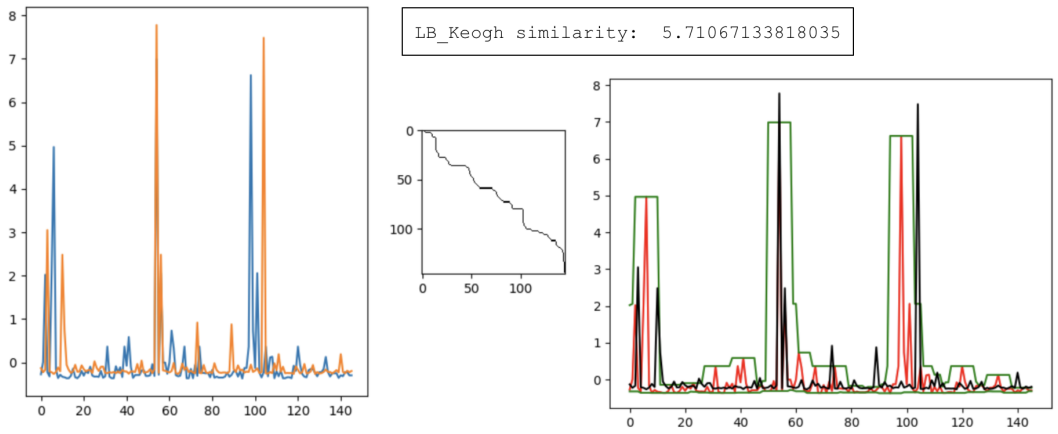

Figure 2.19: LB Keogh similarity (lower bound [25])

while the profile index contains the index of the first neighbor. In other words, it is

the position of the most similar subsequence.

Specifically, a motif is defined as a repeated pattern in a time series, and a discordance is an anomaly. With the matrix profile calculated, it is easy to find the top-K number of patterns or discordances. Which means that a distance close to 0 is most similar to another sub-sequence in the time series and a distance away from 0, say 50, is different from any other sub-sequence. Extracting the smallest distances gives the patterns and the largest distances the discordances.

Chapter 3

Methods

3.1 Supervised fraud analysis

Supervised learning (SL) is the machine learning activity of learning a function that associates an input with an output based on pairs of input-output examples. Specifically, it is able to infer a function from training data labeled by a set of training examples. Each example, in supervised learning, is a pair consisting of an input object (typically a vector) and a desired output value (also called a supervisory signal); in the case of this analysis, each example is a time series representing an account or all transactions of a given user.

The fraud label was used in the supervised learning phase to train our model to classify users as fraudulent or non-fraudulent. To make the supervised predictions, k neighbors time series classifiers were trained and compared. It must be noted that the dataset under analysis fraudulent users are about 0.01% of the total transactions. Given the strong unbalance of the data in this division, it is advisable to use special libraries of "unbalanced learning" able to increase the least represented class (in this case fraudulent) with mathematical techniques to improve learning.

3.1.1 Model: KNN

We used a supervised prediction model K Neighbors Time Series Classifier for which different scores were evaluated using the following pipeline:

1. Normalize the dataset: in our case study we used the time series scaler from the tsleanr preprocessing library. Specifically, the time series scaler is able to scale the time series so that their interval remains between two thresholds (minimum and maximum), in our case we used the standard interval between 0 and 1.

2. K Neighbors Time Series Classifier: Classifier of the tslearn library that implements the k-nearest neighbors vote for time series.

This pipeline has been used for Grid Search CV of the sklearn library which allows to optimize the number of n neighbors used by our classifier. In addition, it was appropriate to carry out different analyses to vary the range of transactions under analysis. Varying the number of transactions for each user it was in fact appropriate to evaluate the time series classifier on a finite number of transactions (20, 25, 30, etc. transactions) in order to compare as many users as possible. Two different weights were also evaluated:

1. Uniform: uniform weights. All points in each neighborhood are weighted equally.

2. Distance: weights the points according to the inverse of their distance. In this case, the neighbors closest to a query point will have a greater influence than the most distant neighbors.

K Neighbors Time Series Classifier can use different metrics, we have focused on the default one, DTW which is the object of our studies for the analysis of time series.

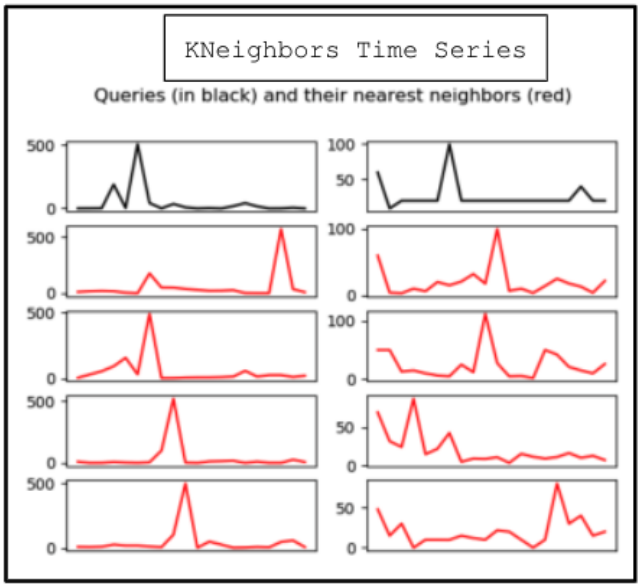

Figure 3.1: Knn Search

In figure 3.1 is represented an example of two time series of users (in black) and its closest users (in red). Through similar user characteristics we can learn significant behaviors for the user under analysis. This method of research could be

used, during the computation of the the financial risk, to assign a weight dependent on the distance from the closest fraudulent user.

Additional models

The following models were used as a comparison [28]: Decision tree [29], bagging [30], balanced bagging [31], random forest [32], balanced random forest [33], easy ensemble [34], RUSBoost [35]. Specifically, the methods were applied on a finite number of transactions (25 transactions) assuming that fraud was to be identified within a finite number of transactions. Specifically, users were used as different samples while time series values were used as features to characterize the samples. In these supervised methods the label used to tow was label fraud. Accuracy over the different models was calculated by taking into account the imbalance of fraudulent versus non-fraudulent users and then weighted appropriately over the two classes by favoring predictions about fraudulent users. (In smaller numbers by orders of magnitude)

3.2 Time series clustering

In this analysis, time series clustering is used for two separate analyses. One to separate users into clusters of membership similar to each other i.e., having similar time series. One of pattern recognition (i.e. user behaviours clustering) to identify significant behaviors or series of transactions for the purpose of reconstructing the user's time series.

3.2.1 Users clustering approaches

Both the two groups of users, fraudolent and non-fraudolent, are here divided into 3 clusters. All the users belonging to the same cluster have a similar spending behaviour.

We represent users with time series and we divide the users into cluster using k-means and k-shape unsupervised time series clustering methods. K-shape algorithm is particularly suitable for univariate time series analysis, as it preserve the shapes of the time series and is invariant to scaling and shifting ([36], [37]).

The K-shape centroid represents the spending behaviour of users belonging to the same cluster. In figure 3.2 are shown the 3 centroids obtained for the non-fraudolent users (upper image) and for the fraudolent users.

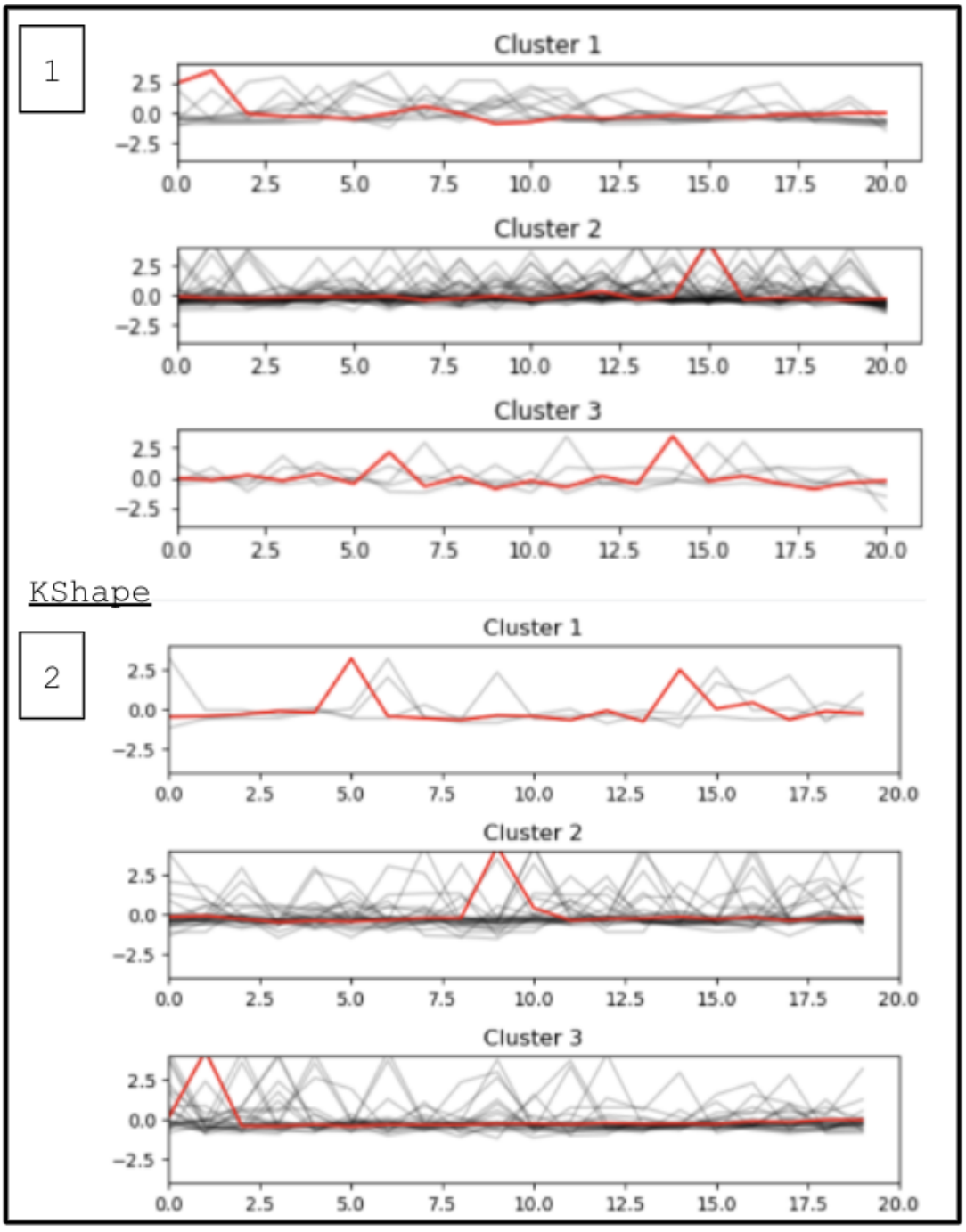

Figure 3.2: KShape centroids for the 3 clusters of non-fraudolent users (1) and fraudolent users (2). [37]

Downsampling and quantization

Before applying the unsupervised clustering methods, the time series are transformed using:

- Piecewise Aggregate Approximation (PAA). PAA corresponds to a downsampling of the original time series and, in each segment, the mean value is retained.
- Symbolic Aggregate approXimation (SAX). SAX builds upon PAA by quantizing the mean value.
- 1d-SAX. It is an extension of SAX in which each segment is represented by an affine function (2 parameters per segment are hence quantized: slope and mean value).

Figure 3.3 shows an example of results obteneid applying the above methods on a fraudolent time series and a non fraudolent time series. This should "smooth out" the time series by removing some less relevant details. This procedure can be particularly efficient in the case of large data sets together with proper indexing ([38]).

Centroids

The centroid of a cluster, obtained with either Kmeans or Kshape, can have the following definitions:

- Euclidian
- DBA
- Soft-DTW

In figure 3.4, 3 methods for calculating the centroid of time series are shown. The centroid is descriptive of the behavior of the users belonging to the same cluster.

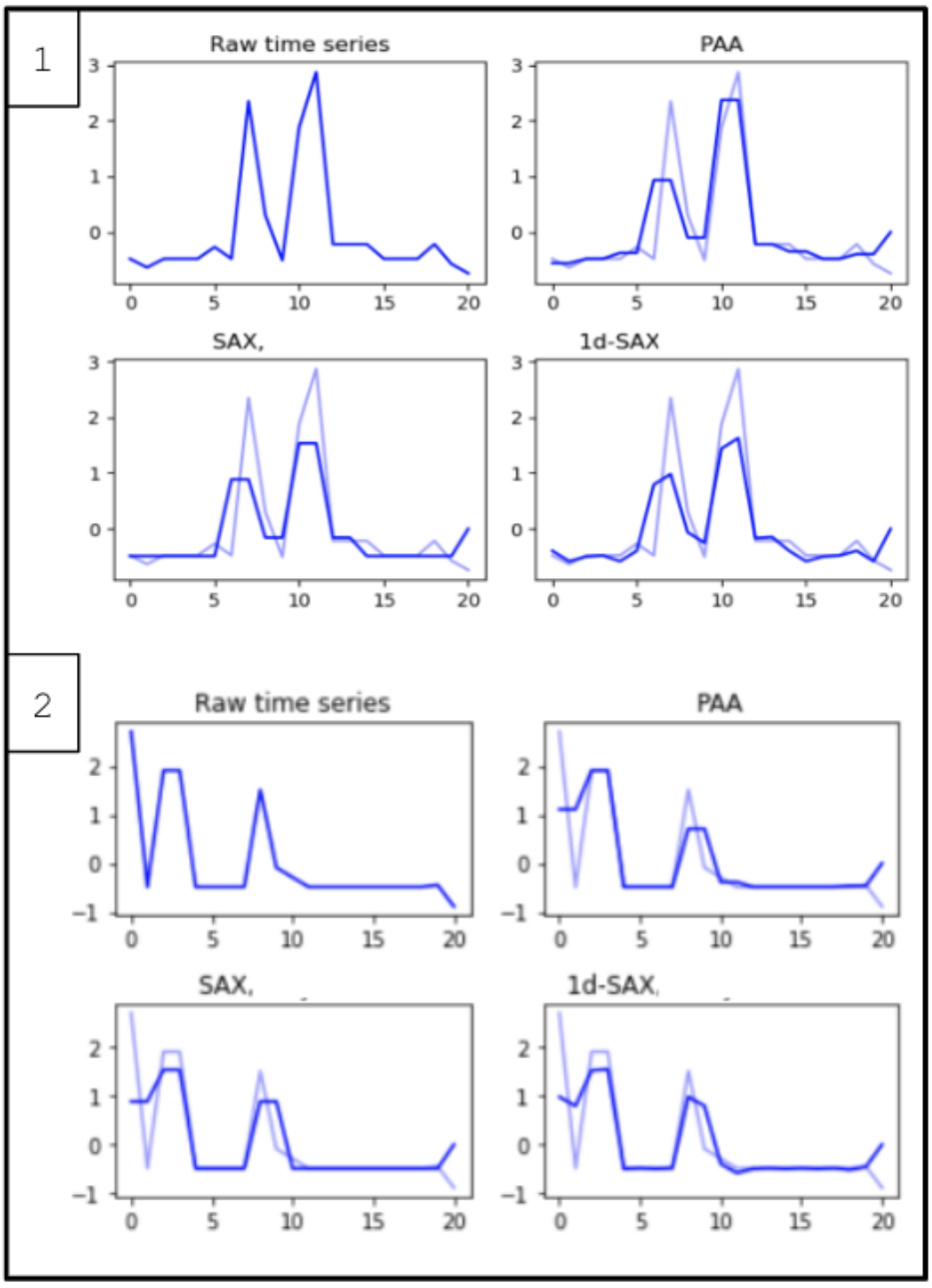

Figure 3.3: Example of time series downsampling (PAA) and quantization (SAX and 1d-SAX)

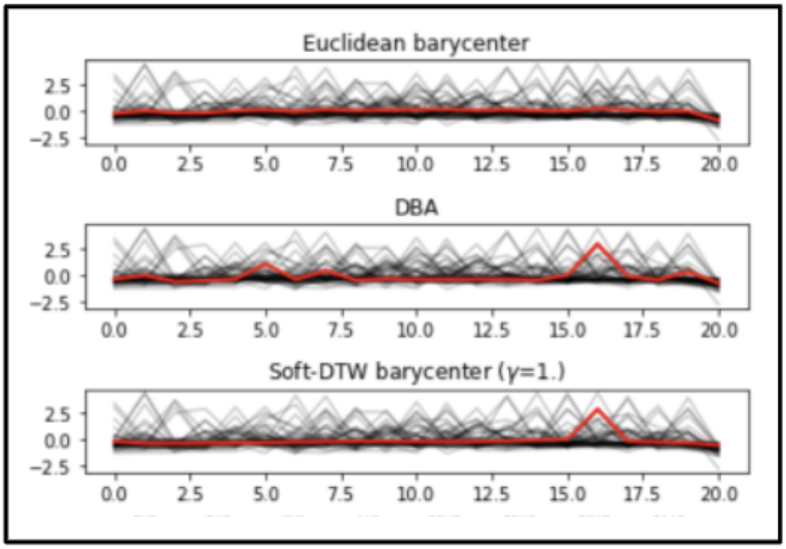

Figure 3.4: Example of KMeans centroids caltulated using Euclidian, DBA, Soft-DTW

3.2.2 User behaviour clustering

In the previous phase the users were divided into clusters. Now, the idea is to characterize the clusters obtained by individually analysis of users belonging to the same cluster.

The adopted unsupervised analyses concern the search for behaviors (or shapeles) that describe portions of time series. As introduced in the chapter of data representation in 2.3.2. We are going to divide the time series into time windows and in addition to the matrix profile we are going to apply the KMeans method to evaluate the reconstruction error once the most significant time windows are clustered. Similar to [39] and others in the literature, we can divide them into more or less broad windows that allow us to analyze the time series also thanks to normalization tools. In the image 3.5 we see a portion of example showing the first 3 windows of a timeseries formed by 58 segments, obtainable through a simple algorithm as reported by the code:

Listing 3.1: Segment division for window analysis as implemented in [2]

```
    segments = []
    for start_pos in range(0, len(time_serie), slide_len):
        end_pos = start_pos + segment_len
        segment = np.copy(time_serie[start_pos:end_pos])
        # Truncating the last incomplete window for simplicity
        if len(segment) != segment_len:
            continue
```

```
8            segments.append(segment)
9
```

In the examples, analyses using a segment_len = 32 (i.e., windows of 32 transactions) and slide_len = 2 (shifted by 2 transactions each) were illustrated and compared

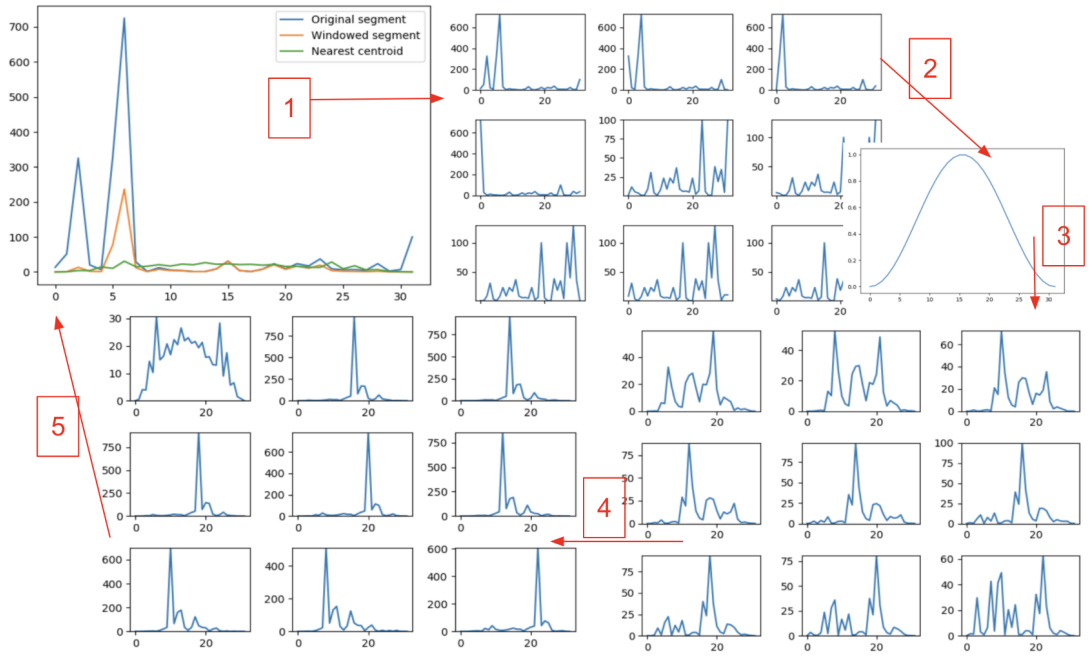

Figure 3.5: Flow of division in windows, normalization, clustering and reconstruction on the transactions of a single user

In particular in figure 3.5 we see how from the initial time series, in blue, we can divide with the above algorithm into windows, step [1]. These windows are properly normalized, step [2], obtaining more comparable and clusterizable time series, step [3]. These windows can be clustered with the KMeans cluster, whose example output is shown in step [4]. At this point the starting segment can also be reconstructed with step [5] to check if the procedure is meaningful and to obtain a time series with the information extracted in the clustering process.

The same clustering method in figure 3.5 was also applied to ordered time series considering increasing amount and considering the label guessed category.

In particular, in Figure 3.6 we see how this clustering method was applied and compared to the time series (in blue) sorted with the category label (guessed_category), step [1]. These windows are appropriately normalized, step [2], resulting in more comparable and clusterizable time series, step [3]. These windows are clustered with KMeans, the example output of which is shown in step [4]. At this point

the starting segment can be reconstructed with step [5] to check if the procedure is meaningful and to obtain a time series with the information extracted in the clustering process.

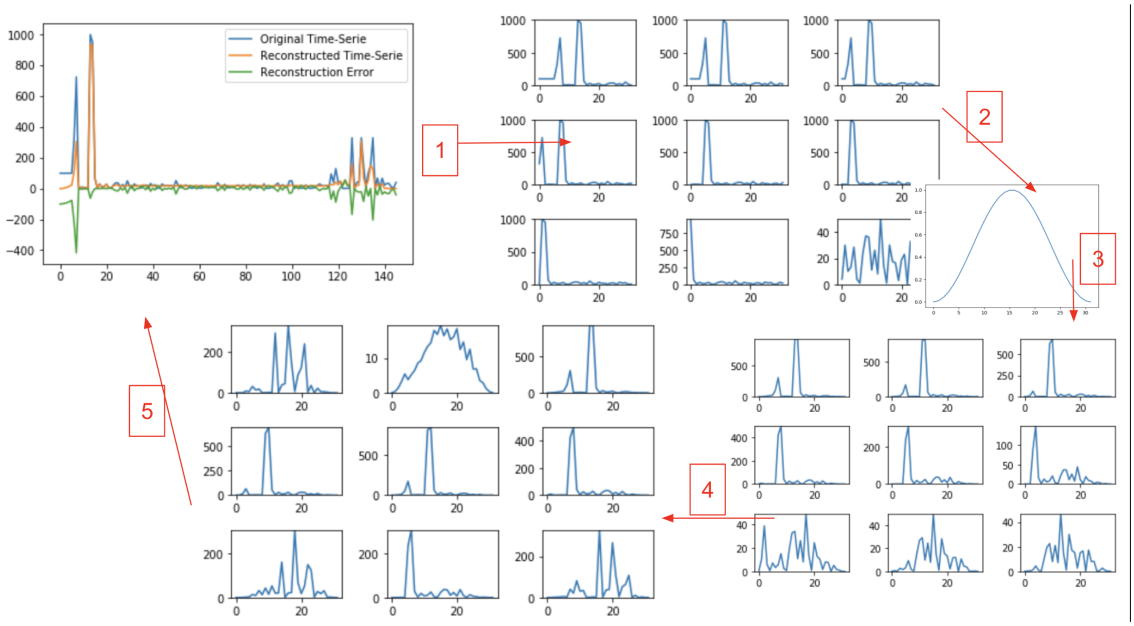

Figure 3.6: Flow of division in windows, normalization, clustering and reconstruction on the transactions of a single user ordered by guessed category label

The same steps are repeated in figure 3.7 applied to the time series sorted by increasing amount.

3.3 Time series forecasting

Time series forecasting is done when one wants to make scientific predictions based on historical data. This forecasting process brings the construction of models generated from historical analysis to make future observations and guide strategic decision making. In our case, several prediction models were used for each individual user. This will allow to predict the credit score in future time windows and to extract useful statistics (seasonality, stationarity, trend, residual) to distribute the credit score over the whole time series of transactions.

As introduced in section 2.3.1, the common libraries for time series forecasting calculate the components of the time series using properties such as Seasonal Mean, Standard Deviation and Autocorrelation. An example is represented in figure 3.8. In the first graph at the top you can see the original time series in gray, seasonal mean in red and standard deviation in blue. This allows you to

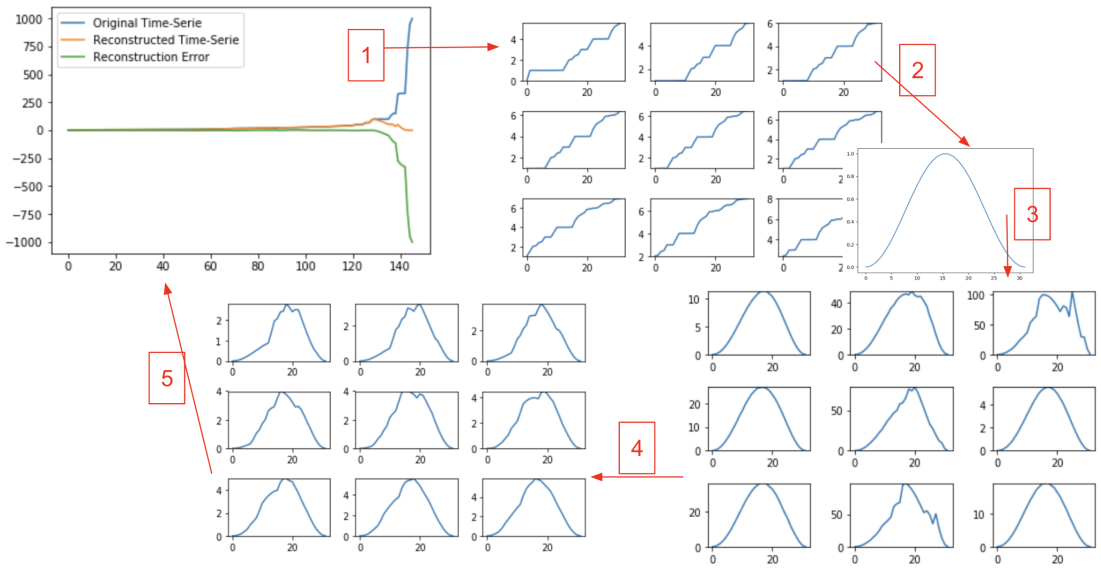

Figure 3.7: Flow of division in windows, normalization, clustering and reconstruction on the transactions of a single user ordered by amount ascending

observe the time series without focusing on individual transactions but evaluating the overall trend over time. In the second graph of the figure 3.8. we evaluate the correlation between individual transactions based on the amount to store and reuse this information where appropriate. In general, we note that it is difficult to have significant auto-relationships if not for certain accounts and therefore this information can be used only in the presence of domain experts.

3.3.1 Forecasting models

The models used are those popular in the literature: autoregressive integrated moving average (ARIMA) model, Autoregression (AR) model, Autoregressive integrated Moving Average (ARMA) model, AutoArima model, Seasonal Autoregressive Integrated Moving-Average (SARIMA) model, Auto Sarima model.

These forecasts were also compared with results obtained from more recent forecasting tools such as Prophet (from Facebook). In particular, thanks to these libraries for the prediction of time series, by temporally dividing the user's time series into a train portion and a test portion, it is possible to calculate an error on the prediction and then use this information by properly evaluating these models; in fact, the results show that there are users who are "easy" to predict while others present more "unpredictable" operations.

These are forecasts made on individual users, so the forecast error remains related to the individual user and therefore can only be included in his risk score

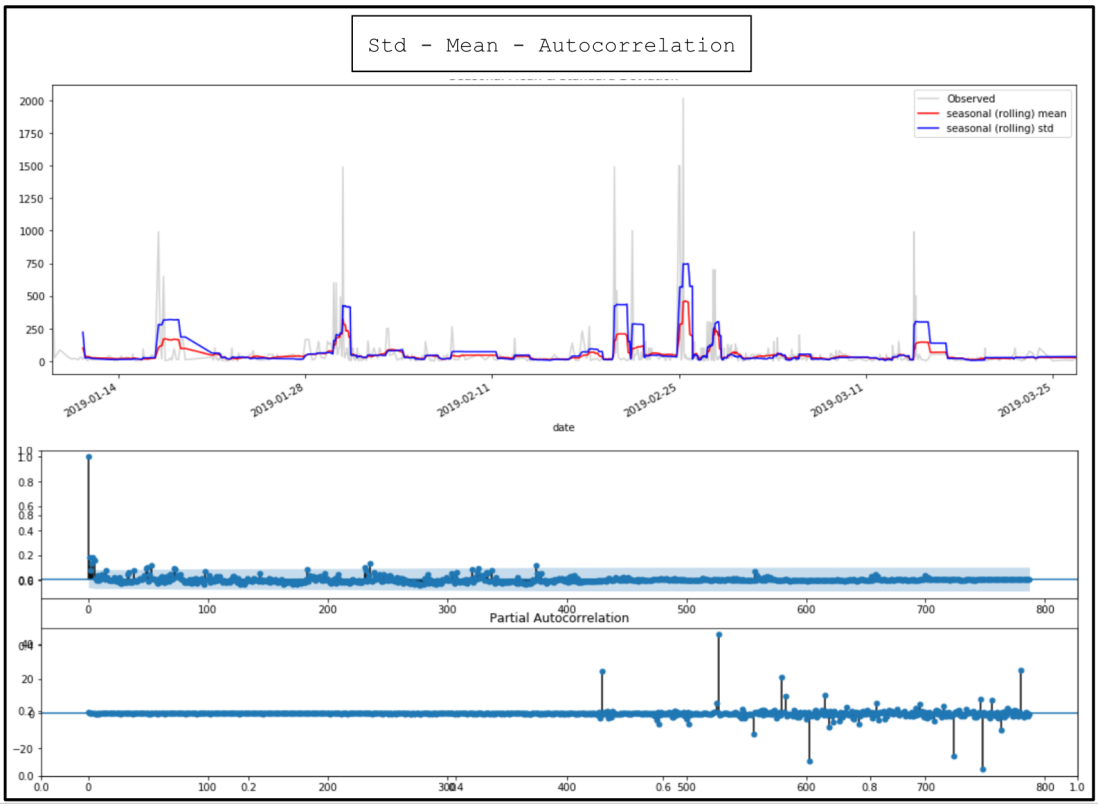

Figure 3.8: Top image: example of time series with mean Standard Deviation values.Bottom image: Auto-Correlation and Partial-Correlation calculated for the same time series

estimate if it is sufficiently accurate.

An example of the division of a time series in train and test set is shown in Figure 3.9. The original time series of the user under analysis has transactions during 3 months (January, February, and March 2019). In the bottom image, the time series was divided into train and test by considering a time division of 2 months (January and February) for train (blue segment) and 1 month (March) for test (orange segment after the dashed vertical line).

Evaluation of forecasting models

The following 4 parameters were used to evaluate the results on the test set:

1. MAE: mean absolute error, calculated as the average of the forecast error values, where all of the forecast error values are absolute

2. MSE: mean squared error, calculated as the average of the squared forecast

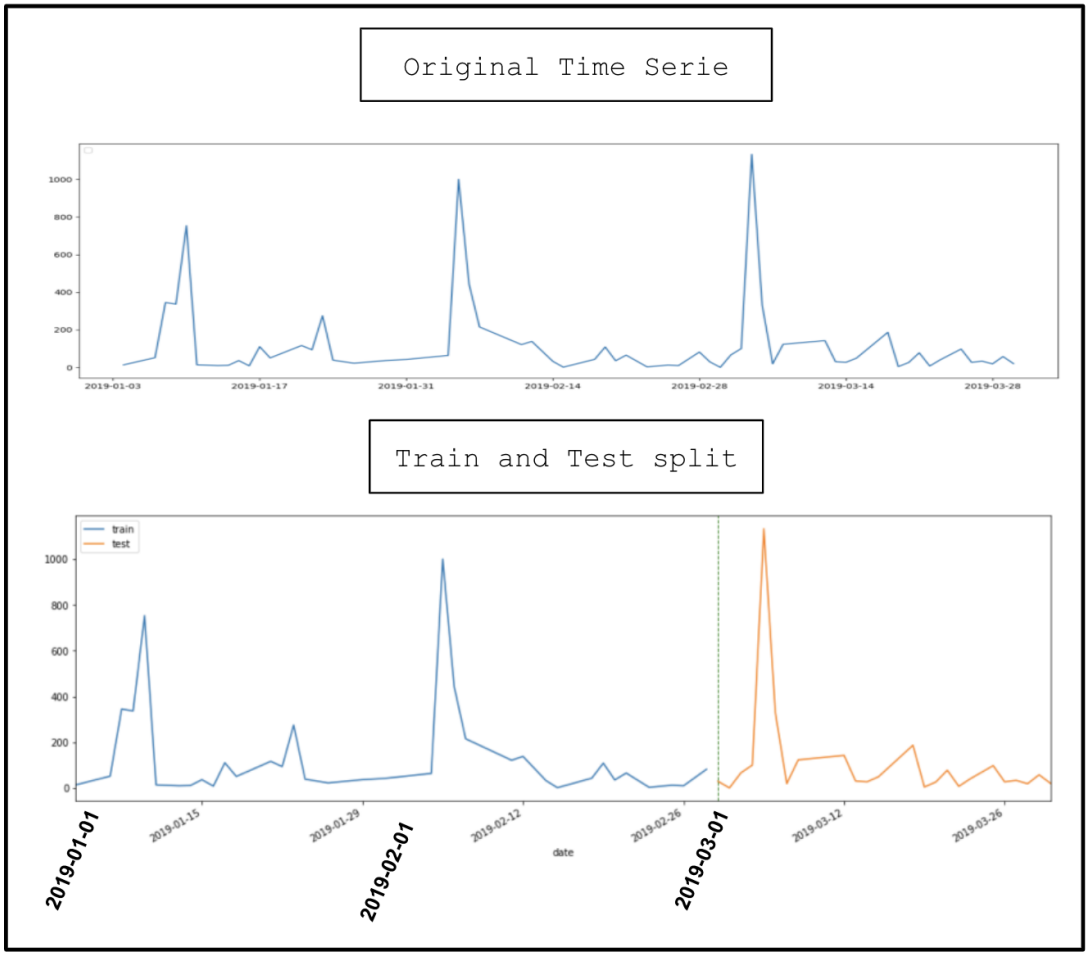

Figure 3.9: Top image: representation of the original time series (over 3 months of transactions); Bottom image: representation of the time series divided into train (first two months, January and February) and test (third month, March)

error values. Squaring the forecast error values forces them to be positive; it also has the effect of putting more weight on large errors.

3. R2: R-Squared or the coefficient of determination, a statistical measure in a regression model that determines the proportion of variance in the dependent variable that can be explained by the independent variable

4. U: Theil's U statistic is a relative accuracy measure that compares the forecasted results with the results of forecasting with minimal historical data.

3.4 Technologies

SC, in addition to providing a sample of anonymized data, has allowed the study to be performed on a proprietary PC and within the protected company network. Each machine learning algorithm and technique used is based on local data computation and confidentiality guaranteed by the use of open source libraries.

3.4.1 Hardware

All methods and analyses were performed only on the computer authorized by the partner company. The computer is described by the following hardware characteristics:

1. **Model:** MacBook Pro (13-inch, 2018, Four Thunderbolt 3 Ports)
2. **Processor:** 2,3 GHz Intel Core i5 quad-core
3. **Memory:** 16 GB 2133 MHz LPDDR3
4. Graphics card: Intel Iris Plus Graphics 655 1536 MB

3.4.2 Software

The following softwares have been installed:

1. Anaconda Navigator 2.0.4
2. Visual Studio Code 1.67.2
3. https://www.python.org/ **Python 3.8.8**
4. https://dtaidistance.readthedocs.io/en/latest/ **DTW, Clustering**
5. https://scipy.org/ **Scipy: stats**
6. https://numpy.org/ **Numpy**
7. https://matplotlib.org/ **Matplotlib**
8. https://tslearn.readthedocs.io/ **Tslearn: preprocessing, barycenters, svm, clustering, piecewise, neighbors, utils, model selection, pipeline, metrics**
9. https://github.com/pandas-profiling/pandas-profiling **Pandas Profiling**
10. https://scikit-learn.org/ **Sklearn: KMeans**

11. https://imbalanced-learn.org/dev/references/generated/imblearn.under_sampling.NearMiss.html **Imbalanced learn: NearMiss**
12. https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html **Sklearn: svm LinearSVC**
13. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html **Sklearn: DecisionTreeClassifier**
14. https://imbalanced-learn.org/stable/references/generated/imblearn.ensemble.BalancedBaggingClassifier.html **Imbalanced learn: BalancedBaggingClassifier**
15. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html **Sklearn: BaggingClassifier**
16. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html **Sklearn: RandomForestClassifier**
17. https://imbalanced-learn.org/stable/references/generated/imblearn.ensemble.BalancedRandomForestClassifier.html **Imbalanced learn: BalancedRandomForestClassifier**
18. https://imbalanced-learn.org/stable/references/generated/imblearn.ensemble.EasyEnsembleClassifier.html **Imbalanced learn: EasyEnsembleClassifier**
19. https://imbalanced-learn.org/stable/references/generated/imblearn.ensemble.RUSBoostClassifier.html **Imbalanced learn: RUSBoostClassifier**

Chapter 4

Results & Discussion

4.1 Supervised fraud analysis

In this chapter the results of the supervised learning phase are shown and discussed. As explained in 3.1, different machine learning model (KNN, Decision tree, bagging, balanced bagging, random forest, balanced random forest, easy ensemble, RUSBoost) were trained to classify the users in two groups: fraudulent and non fraudulent. The ground truth data to train the models (fraud label) was provided by the company. We can consider this information reliable as it has been provided and validated by legal experts in the financial field.

4.1.1 Results

The data, i.e., time series, are represented as described in the 2.3 section, i.e., each user is described by transaction values over time. In other words, each user time series x is a row in the matrix X of all users analyzed. The last column of X are the values of the "Fraud" label to be predicted. The number of fraudulent users is only 34 as there are only 34 users with minimun 20 transactions, while the non fraudulent users are 604.

Regarding the division into train set and test set, the dataset is divided ensuring the stratify property using a standard test_size = 0.25 (25%). Given the strongly unbalance of the dataset, we will refer to this case as "**unbalanced**".

In fact, to ensure that the number of fraudulent user is equal to the number of non fraudulent users, a second split was made on a smaller balanced dataset. We will refer at this case a "**balanced**". The smaller dataset contains 34 fraudulent users and 34 non-fraudulent.

To perform the tuning of the hyperparameters of the KNN classifier, a pipeline consisting of two steps was used. First, the data are normalized using min-max normalization. Next, they are sent to a KNN classifier. For the KNN classifier,

the n_neighbors and weights hyperparameters are tuned using the cross validation approach. Specifically, n_split = 2 was used for the StratifiedKFold with an initial shuffle and seed = 42; for the KNN classifier, models were constructed with k_nn (number of neighbors) = 5 and 25 with different weights (uniform, distance) and with different metrics (DTW, L2, SAX+MINDIST)

KNN

Tables 4.1 and 4.2 shows the results of the proposed model (KNN - k nearest neighbors time series classifier) in two cases, unbalanced on table 4.1 and balanced on table 4.2, comparing the first 20 transactions, on all users. The tables represent the users in the two cases, balanced and unbalanced. A significant problem in fact is that fraudulent users have few transactions and often the fraudulent user manages to create more users with few transactions, so it is more difficult to be identified by any model. Tables show results for different n neighbors (5, 25), different knn weights (uniform, distance) for two different folds.

n neighbors	knn weights	score fold 1	score fold 2
5	**uniform**	**0.99**	**0.99**
5	**distance**	**0.99**	**0.99**
25	**uniform**	**0.99**	**0.99**
25	**distance**	**0.99**	**0.99**

Table 4.1: Results of knn with different n neighbors (5, 25) and different knn weights (uniform, distance); **unbalanced case**

n neighbors	knn weights	score fold 1	score fold 2
5	**uniform**	**0.53**	**0.56**
5	**distance**	**0.53**	**0.56**
25	**uniform**	**0.53**	**0.50**
25	**distance**	**0.53**	**0.53**

Table 4.2: Results of knn with different n neighbors (5, 25) and different knn weights (uniform, distance); **balanced case**

Table 4.3 shows the results of the k neighbors time series classifier in the two cases, the unbalanced case (with all users) and the case balanced 50% by fraudulent users and 50% by non-fraudulent users. These values were obtained using n neighbors = 3.

case	DTW	L2	SAX+MINDIST(L2)
unbalanced	**0.99**	**0.98**	**0.99**
balanced	**0.59**	**0.71**	**0.65**

Table 4.3: Nearest neighbor classification comparison using DTW, L2 and SAX (Symbolic Aggregate Approximation)

Comparison with other classifiers applied to time series

The models parameters are the default proposed by the scikit-learn library ([40]). The number of estimators was set to 50 for Bagging and balanced Bagging,Random forest and balanced random forest; the numbers of estimators for AdaBoostClassifier [41], Easy ensemble and RUSBoost was set to 10.

In Figure 4.1 where the results obtained for 604 non-fraudulent users and 9 fraudulent users are compared since they had at least 25 transactions that had passed the prerequisites of section 2.2, i.e. 25 transactions considered valid for various models, starting from the left Decision Tree Classifier, Bagging classifier and Balanced bagging classifier with the following scores:

Classifier	Balanced accuracy	Geometric mean
Decision tree	**0.55**	**0.33**
Bagging	**0.50**	**0.00**
Balanced bagging	**0.74**	**0.72**

Table 4.4: Results of Decision tree, bagging and balanced bagging classifiers

In Figure 4.2 where the results obtained for 604 non-fraudulent users and 9 fraudulent users are compared since they had at least 25 transactions that had passed the prerequisites of section 2.2, i.e. 25 transactions considered valid for various models, starting from the left Random Forest Classifier, Balanced Random Forest, Easy Ensemble classifier and RUSBoost classifier with the following scores:

4.1.2 Discussion

The label Fraud subject of the supervised prediction was manually assigned by personnel capable of recognizing fraudster or fraud attempts and was provided with the dataset under analysis thanks to the collaboration with the partner company to the thesis. Therefore, we can consider this information reliable as it has been provided and validated by legal experts in the financial field.

Determining the accuracy of the learned feature in general depends strongly on how the input object is represented. In this dataset, which for reasons of

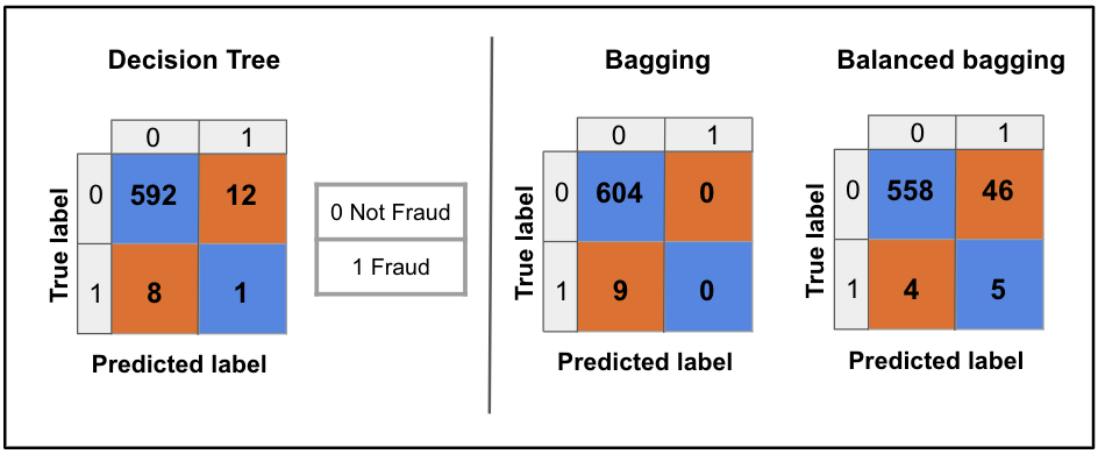

Figure 4.1: Comparison of classifiers on a subset of users having at least 25 transactions, which are 604 non-fraudulent and 9 fraudulent users. On the left side you can see the Decision Tree results, on the right side Bagging and Balanced bagging classifiers.

Classifier	Balanced accuracy	Geometric mean
Random forest	**0.50**	**0.00**
Balanced random forest	**0.74**	**0.74**
Easy ensemble	**0.69**	**0.69**
RUSBoost	**0.75**	**0.75**

Table 4.5: Results of random forest, balanced random forest, easy ensemble and RUSBoost classifiers

anonymization and the inherent nature of the PSD2 system, has lost much of its information, it is not easy to estimate accuracy.

Notably, multiple analyses were conducted to compensate for the problem of imbalance in fraudulent data across various user groups. Given the low scores obtained, less than 80 % accuracy, prediction models were also conducted on certain ranges of transactions by filtering transactions between a minimum and maximum threshold but this increased accuracy only slightly. Having reached a value of not satisfactory precision, these supervised analyses will be considered only in minimal measure in the evaluation of the risk in the score of the user.

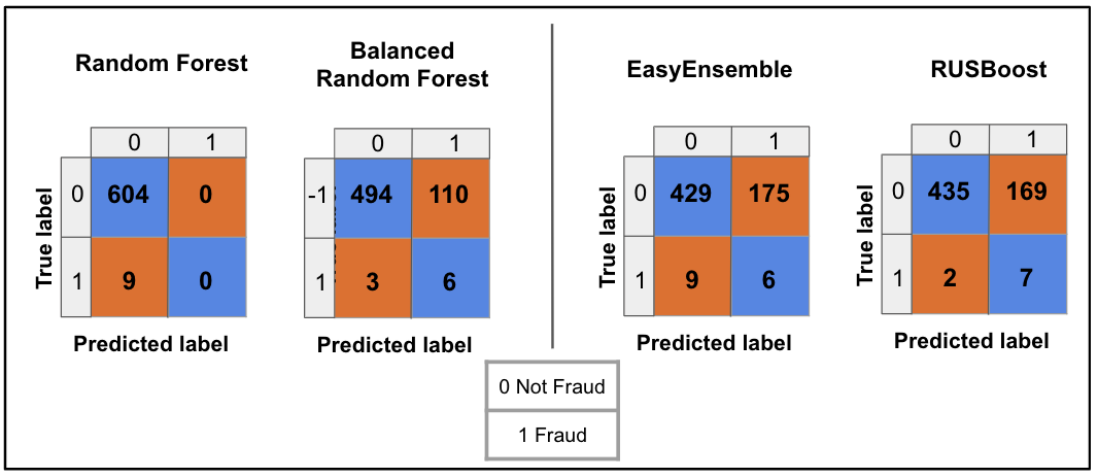

Figure 4.2: Comparison of classifiers on a subset of users having at least 25 transactions, which are 604 non-fraudulent and 9 fraudulent users. On the left side you can see the Random Forest results in its unbalanced and balanced form, on the right side EasyEnsemble and RUSBoost balanced classifiers.

4.2 Time series clustering

In this section the outcome of the unsupervised learning phase are discussed. The proposed clustering models were scaled using 'TimeSeriesScalerMeanVariance' Scaler for time series. That is, the time series were scaled so that their mean (or standard deviation) in each dimension is mu (or std) with mu=0.0 and std=1.0. As explained in 3.2, clustering is used for two separate analyses:

- Users Clustering.

  One to separate users into clusters of membership similar to each other i.e., having similar time series. All individuals (i.e all the time series) within the same cluster have the same spending behaviour. Upon confirmation by experts, the same credit score should be assigned to users belonging to the same cluster. In fact, it must be noted that the company didn't provide the users credit risk score and therefore we cannot confirm the hypothesis that users belonging to the same cluster have the same credit score.

- User behaviour clustering. One of pattern recognition (i.e. user behaviours clustering) to identify significant behaviors or series of transactions for the purpose of reconstructing the user's time series. To select the appropriate number of clusters of each behavior, the number of clusters that reduced the reconstruction error was used. Using the clusters with lower reconstruction

error it is possible to assume that these behaviors are significant, however they should be subjected to analysis by experts in finance and econometrics.

Users clustering

Unsupervised learning algorithms are used to find recurring or significant behaviors in the training dataset, i.e., common transactions or spending patterns. Therefore, clustering techniques are used in which the algorithm automatically groups the training records into categories with similar characteristics, categorizing wallets and users. As in the case of k-shape also in the case of the k-means kernel the clusters produced depend strongly on the order of the time series of trains. This also depends strongly on the type of data we are providing, in fact as we increase the number of minimum transactions owned by the series, the better the clustering will be, increasing the degree of differentiation of the series. With the data in our possession it is difficult to evaluate the effectiveness of clustering, in fact, by increasing the number of transactions per series we reduce the number of users who meet this condition.
The cluster obtained can not be validated as we miss the ground truth, which must be provided by domain experts. Future works should analyze how the centroids of the cluster can influence the credit score. Users belonging to the same cluster should be therefore assigned with a similar credit score.

User behaviour clustering: time series reconstruction error

User behaviour clustering was applied to a user's time series. It was also applied to the same users but by ordering the time series considering increasing amount and considering the label guessed category. This means that the transactions are not ordered in time. Figure 4.3 compares the various errors in the 3 cases discussed. In particular, the maximum reconstruction error and the 98th percentile for the reconstruction error were analyzed. The calculated value on the amount error, on the y-axis, varying the number of clusters, x-axis, is compared. Specifically in the same figure 4.3, graph [1] refers to the original time series analyzed in figure 3.5; graph [2] refers to the time series of the same user sorted by label category analyzed in figure 3.6; graph [3] refers to the time series of the same user sorted by increasing amount analyzed in figure 3.7. It is possible to observe that the best results (i.e. the minimum number of cluster needed to reconstruct the time series) were obtained by sorting the time series by their amount.

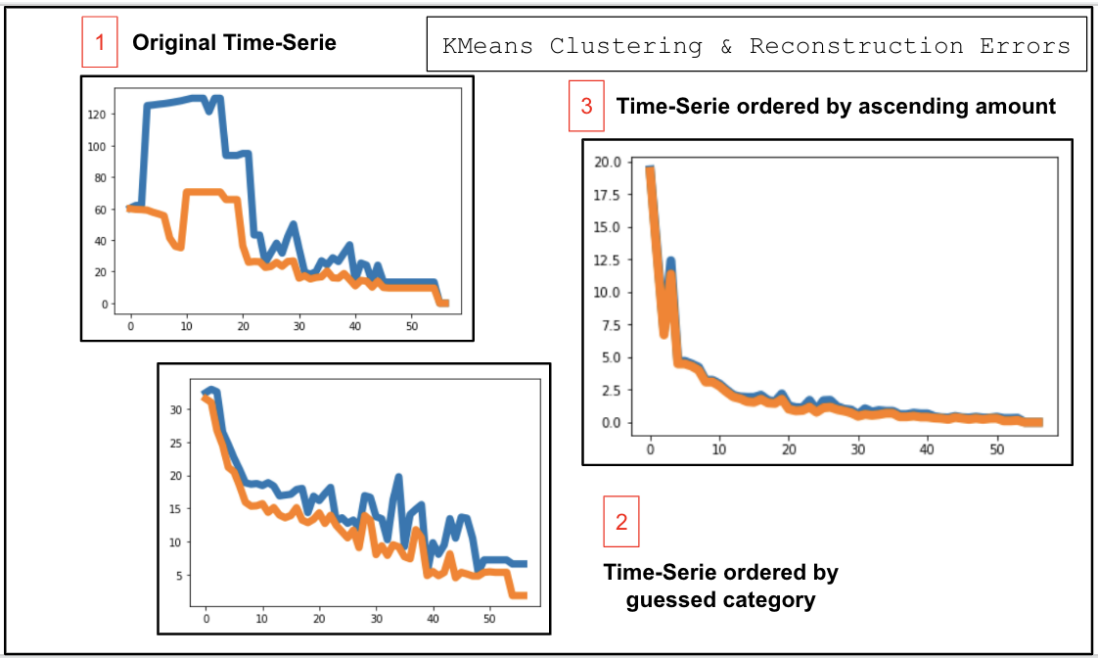

Figure 4.3: Representation of maximum reconstruction error and 98th percentile of reconstruction error. The calculated value on the amount error, y-axis, varying the number of clusters, x-axis. The graph [1] refers to the original time series analyzed in figure 3.5; the graph [2] refers to the time series of the same user sorted by label category analyzed in figure 3.6; the graph [3] refers to the time series of the same user sorted by increasing amount analyzed in figure 3.7.

4.3 Time series forecasting

In this chapter ARIMA and PROPHET forecasting methods are evaluated and compared using the following metrics: RMSE, MSE, R2, U as explained in chapter 3.3.1. auto_ARIMA tuning process was used to find ARIMA optimal differencing parameter d [42]. Since PROPHET method detected a weekly seasonality of the data, the parameter seasons = 7 was set for ARIMA.

4.3.1 Results

Figure 4.4 shows the mean performances of the two forecasting methods on all the users. For each user, each method was trained using the first 2 months of transactions and tested on the third months.

The metrics express how good the methods predict the third month of transactions. According to the metrics, Prophet has better performances (smaller RMSE, MSE and R2 and higher U).It must be noted the the mean values of the transactions

is 834 EUR and the mean standard deviations values is 1576 EUR.

	ARIMA	PROPHET
RMSE (EUR)	1452	1192
MSE (EUR)	11'263'308	6'510'464
R^2	-0.45	-0.04
U	1.28×10^{-3}	1.57×10^{-3}

Figure 4.4: Prophet and ARIMA forecasting methods performances

Figure 4.5 shows a clear example for which prophet performs better than ARIMA (the metrics related to this example are reported in Figure 4.6

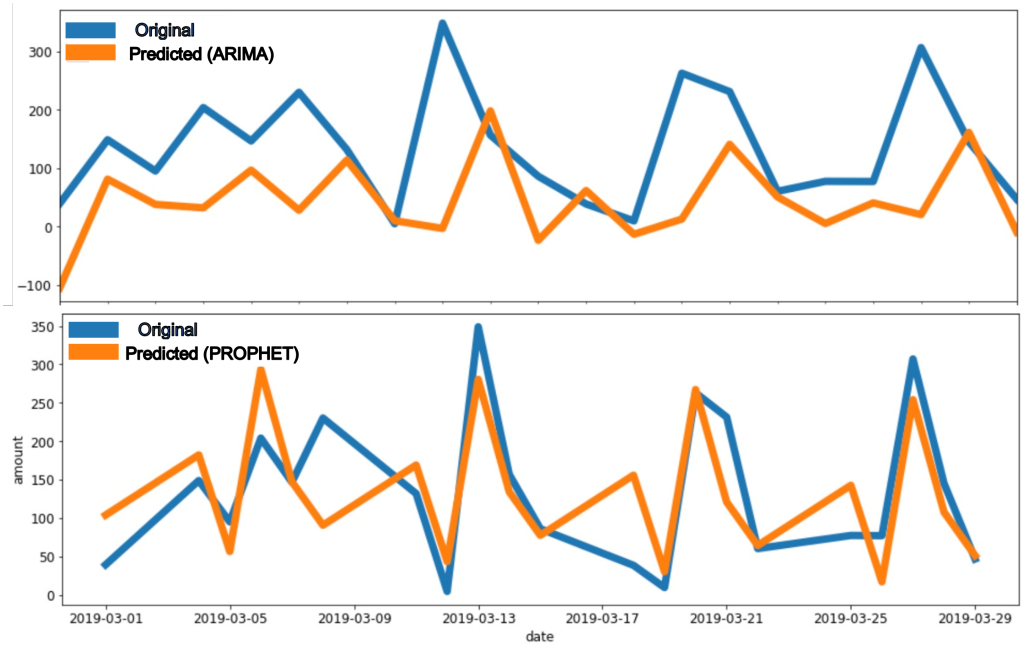

Figure 4.5: Prophet and ARIMA predictions on a single user. PROPHET has better performances.

	Models	RMSE Errors	MSE Errors	U Errors	R2 Errors
0	ARIMA	138.610759	19212.942533	0.010288	-1.118166
1	Prophet	61.986709	3842.352049	0.002391	0.576393

Figure 4.6: Prophet and ARIMA performances on a single user. PROPHET has better performances.

4.3.2 Discussion

The results obtained are not satisfactory. Prophet seems to perform better than ARIMA, as it more suited for seasonal data. It is not possible to assume that the models are descriptive for all users, and it would be appropriate for domain experts to evaluate for each user and for each time series which model is most appropriate and useful for the case study.

Chapter 5

Conclusions

The collected results reveal how complex and competitive it is to find valid and complete solutions to correctly estimate a representative probability of default risk.

In the first place, it is evident that fraudulent individuals are able to implement complex schemes that are difficult to automate and identify. In many situations, fraudulent individuals are able to generate multiple accounts and users and use them only for certain transactions, consequently making any time series approach useless. Furthermore, in the context of microtransactions and online transactions in which we live today, transactions can easily be modified and take countless forms through countless intermediaries via refund systems and account transfers.

The main criticism in not being able to collect satisfactory results was in fact the impossibility of using much of the main information related to the transaction and the confidential information of the user. The context of anonymization has been so severe as to remove much of the usable information from current and modern transaction analysis systems, and PSD2 standards themselves do not provide for the possibility of sharing much of the information that is exchanged between banking systems and anti-fraud monitoring systems.

The most interesting results have been in gathering and standardizing the current state of the art implemented through the use of open source libraries using various approaches that would be possible to use in professional contexts, even at the financial level, without having to invest money in licenses or further private risk profiling analysis.

The current state of the art, allows us to easily understand that there is no globally accepted method that has proven to be better than others except on specific datasets that are not particularly significant at the level of structured research in the field. Hence the importance of further analysis in this area by comparing different machine learning techniques for credit risk analysis. The results obtained from the various calculations are not fully comparable with the results present in the state of the art as they do not contain the same information. In order to extract

additional and financially meaningful information, further analysis by a domain expert would be required to add value to the clusters and predictions presented in this thesis. The same domain experts could validate and classify certain behaviors and/or users as virtuous based on the economic analysis in order to identify the set of users that most closely match those behaviors and propensities. Through these final steps, it would be possible to extend the analyzed methods by providing an additional tool for credit risk score analysis.

Appendix A

Appendix

The following tables A.1, A.2, A.3, A.4, A.5, A.6 provide a list of the articles that were reviewed by [13] classifying them by risk type, including the risk management method/instrument and the algorithms. These researches were taken as examples for the study of models and methods that are implementable to our problem.

Risk Type	Risk Management Method/-Tool	Reference	Algorithm
Compiance Risk Management	Risk Monitoring	Mainelli and Yeandle 2006 [43]	SVM
Compiance Risk Management—Concentration Risk	Stress Testing	Pavlenko and Chernyak 2009 [44]	Bayesian Networks
Credit Risk Management—Consumer Credit	Exposure (PD, LGD, EAD)	Yeh and Lien 2009 [45]	Bayesclassifier, Nearest neighbor, ANN, Classification trees
Credit Risk Management—Consumer Credit	Scoring Models	Bellotti and Crook 2009 [46]	SVM
Credit Risk Management—Consumer Credit	Scoring Models	Galindo and Tamayo 2000 [47]	CART, NN, KNN
Credit Risk Management—Consumer Credit	Scoring Models	Wang et al. 2015 [48]	Lasso logistic regression
Credit Risk Management—Consumer Credit	Scoring Models	Hamori et al. 2018 [49]	Bagging, Random Forest, Boosting
Credit Risk Management—Consumer Credit	Scoring Models	Harris 2013 [50]	SVM
Credit Risk Management—Consumer Credit	Scoring Models	Huang et al. 2007 [51]	SVM

Table A.1: algorithms referenced 1, taken from [13]

Risk Type	Risk Management Method/-Tool	Reference	Algorithm
Credit Risk Management—Consumer Credit	Scoring Models	Keramati and Yousefi 2011[52]	NN, Bayesian Classifier, DA, Logistic Regression, KNN, Decision tree, Survival Analysis, Fuzzy Rule based system, SVM, Hybrid mode
Credit Risk Management—Consumer Credit	Scoring Models	Khandani et al. 2010 [53]	CART
Credit Risk Management—Consumer Credit	Scoring Models	Lai et al. 2006 [54]	SVM
Credit Risk Management—Consumer Credit	Scoring Models	Lessmann et al. 2015 [55]	Multiple algos assessed
Credit Risk Management—Consumer Credit	Scoring Models	Van-Sang and Nguyen 2016 [56]	Deep Learning
Credit Risk Management—Consumer Credit	Scoring Models	Yu et al. 2016 [51]	Deep belief network, Extreme Machine Learning
Credit Risk Management—Consumer Credit	Scoring Models	Wang et al. 2005 [51]	SVM, Fuzzy SVM
Credit Risk Management—Consumer Credit	Scoring Models	Zhou and Wang 2012 [51]	Random Forest

Table A.2: algorithms referenced 2, taken from [13]

Risk Type	Risk Management Method/-Tool	Reference	Algorithm
Credit Risk Management—Coporate Credit	Exposure (PD, LGD, EAD)	Bastos 2014 [57]	Bagging
Credit Risk Management—Coporate Credit	Exposure (PD, LGD, EAD)	Barboza et al. 2017 [58]	Neural Network, SVM, Boosting, Bagging, Random Forest
Credit Risk Management—Coporate Credit	Exposure (PD, LGD, EAD)	Raei et al. 2016 [59]	Neural Networks
Credit Risk Management—Coporate Credit	Exposure (PD, LGD, EAD)	Yang et al. 2011 [60]	SVM
Credit Risk Management—Coporate Credit	Exposure (PD, LGD, EAD)	Yang et al. 2017 [51]	SVR
Credit Risk Management—Coporate Credit	Scoring Models	Ala'Raj and Abbod 2016b [61]	Multiclassifer system (MCS), Ensemble neural networks (NN), support vector machines (SVM), random forests (RF), decision trees (DT) and naïve Bayes (NB).
Credit Risk Management—Coporate Credit	Scoring Models	Ala'raj and Abbod 2016a [62]	GNG, MARS
Credit Risk Management—Coporate Credit	Scoring Models	Bacham and Zhao 2017 [63]	ANN, Random Forest

Table A.3: algorithms referenced 3, taken from [13]

Risk Type	Risk Management Method/-Tool	Reference	Algorithm
Credit Risk Management—Coporate Credit	Scoring Models	Cao et al. 2013 [64]	SVM
Credit Risk Management—Coporate Credit	Scoring Models	Van Gestel et al. 2003 [65]	SVM
Credit Risk Management—Coporate Credit	Scoring Models	Guegan et al. 2018 [66]	Elastic Net, random forest, Boosting, NN
Credit Risk Management—Coporate Credit	Scoring Models	Malhotra and Malhotra 2003 [67]	NN
Credit Risk Management—Coporate Credit	Scoring Models	Wójcicka 2017 [68]	Neural networks
Credit Risk Management—Coporate Credit	Scoring Models	Zhang 2017 [69]	KNN, Random Forest
Credit Risk Management—Coporate Credit	Stress Testing	Blom 2015 [70]	Lasso regression
Credit Risk Management—Coporate Credit	Stress Testing	Chan-Lau 2017 [71]	Lasso regression
Credit Risk Management—Credit Card Risk	Exposure (PD, LGD, EAD)	Yang et al. 2017 [51]	SVM
Credit Risk Management—Cross-risk	Stress Testing	Jacobs 2018 [72]	MARS
Credit Risk Management—Wholesale	Stress Testing	Islam et al. 2013 [73]	Cluster analysis

Table A.4: algorithms referenced 4, taken from [13]

Risk Type	Risk Management Method/-Tool	Reference	Algorithm
Liquidity Risk Management—Liquidity Risk	Risk Limits	Gotoh et al. 2014 [74]	vSVM
Liquidity Risk Management—Liquidity Risk	Risk Monitoring	Sala 2011 [75]	ANN
Liquidity Risk Management—Liquidity Risk	Scoring Models	Tavana et al. 2018 [76]	ANN, Bayesian Networks
Management—Consumer Credit	Scoring Models	Brown and Mues 2012 [77]	Gradient, Boosting, Random Forest, Least Squares—SVM
Market Risk Management—Equity Risk	Value at Risk	Zhang et al. 2017 [78]	Gradient, Boosting, Random Forest, GELM
Market Risk Management—Equity Risk	Value at Risk	Mahdavi-Damghani and Roberts 2017 [79]	Cluster analysis
Market Risk Management—Equity Risk	Value at Risk	Monfared and Enke 2014 [80]	NN
Market Risk Management—Equity Risk	Value at Risk	Kanevski and Timonin 2010 [81]	SOM, Gaussian Mixtures, Cluster Analysis
Operational Risk Management—Cybersecurit	Risk Assessment (RCSA)	Peters et al. 2017 [82]	Non-linear clustering method

Table A.5: algorithms referenced 5, taken from [13]

Risk Type	Risk Management Method/-Tool	Reference	Algorithm
Operational Risk Management—Fraud Risk	Operational Risk Losses	Pun and Lawryshyn 2012 [83]	Neural Networks, k-Nearest Neighbor, Naïve Bayesian, Decision Tree
Operational Risk Management—Fraud Risk	Operational Risk Losses	Sharma and Choudhury 2016 [84]	SOM
Operational Risk Management—Fraud Risk	Risk Monitoring	Ngai et al. 2011 [85]	neural networks, Bayesian belief network, decision trees
Operational Risk Management—Fraud Risk	Risk Monitoring	Sudjianto et al. 2010 [86]	SVM, Classification Trees, Ensemble Learning, CART, C4.5, Bayesian belief networks, HMM
Operational Risk Management—Money Laundering/Financial Crime	Risk Monitoring	Khrestina et al. 2017 [87]	ogistic regression

Table A.6: algorithms referenced 6, taken from [13]

Bibliography

[1] Marco Cuturi and Mathieu Blondel. «Soft-dtw: a differentiable loss function for time-series». In: *International conference on machine learning*. PMLR. 2017, pp. 894–903 (cit. on pp. xii, 31).

[2] Anjin Liu, Jie Lu, and Guangquan Zhang. «Concept drift detection via equal intensity k-means space partitioning». In: *IEEE transactions on cybernetics* 51.6 (2020), pp. 3198–3211 (cit. on pp. xii, 42).

[3] *Financial Risk Score*. `https://twia.msbcommercial.com/Help/en-us/Content/Resources/Glossary/Financial_Risk_Score.htm`. Accessed: 2021-09-30 (cit. on p. 1).

[4] Bart van Liebergen. «Machine learning: A revolution in risk management and compliance?» In: *Journal of Financial Transformation* 45 (2017), pp. 60–67. URL: `https://ideas.repec.org/a/ris/jofitr/1592.html` (cit. on p. 2).

[5] Stefania Pozzuoli. *L'evoluzione del credito alle societÃ non finanziarie e alle famiglie: un'analisi empririca per l'Italia*. Working Papers 2. Department of the Treasury, Ministry of the Economy and of Finance, Feb. 2018. URL: `https://ideas.repec.org/p/itt/wpaper/wp2018-2.html` (cit. on pp. 2, 3).

[6] Ben S Bernanke, Mark Gertler, and Simon Gilchrist. «The financial accelerator in a quantitative business cycle framework». In: *Handbook of macroeconomics* 1 (1999), pp. 1341–1393 (cit. on p. 3).

[7] Suresh Kumar Annappindi. *System and method for predicting consumer credit risk using income risk based credit score*. US Patent 8,799,150. Aug. 2014 (cit. on pp. 3–6).

[8] *FICO Score Definition*. `https://www.investopedia.com/terms/f/ficoscore.asp`. Accessed: 2021-12-05 (cit. on p. 6).

[9] *How are FICO Scores Calculated?* `https://www.myfico.com/credit-education/whats-in-your-credit-score`. Accessed: 2021-12-05 (cit. on p. 6).

[10] Randy Anderson and Christine Elving. *Randy Anderson*. Contemporary Art Gallery, 1986 (cit. on p. 7).

[11] Varsha Aithal and Roshan David Jathanna. «Credit risk assessment using machine learning techniques». In: *International Journal of Innovative Technology and Exploring Engineering* 9.1 (2019), pp. 3482–3486 (cit. on p. 8).

[12] Diederick Van Thiel and Willem Frederik Fred Van Raaij. «Artificial intelligence credit risk prediction: An empirical study of analytical artificial intelligence tools for credit risk prediction in a digital era». In: *Journal of Risk Management in Financial Institutions* 12.3 (2019), pp. 268–286 (cit. on pp. 8, 9).

[13] Martin Leo, Suneel Sharma, and Koilakuntla Maddulety. «Machine learning in banking risk management: A literature review». In: *Risks* 7.1 (2019), p. 29 (cit. on pp. 8, 9, 15, 16, 61–67).

[14] Ellen Tobback and David Martens. «Retail credit scoring using fine-grained payment data». In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 182.4 (2019), pp. 1227–1246 (cit. on p. 8).

[15] Konstantinos Kalpakis, Dhiral Gada, and Vasundhara Puttagunta. «Distance measures for effective clustering of ARIMA time-series». In: *Proceedings 2001 IEEE international conference on data mining*. IEEE. 2001, pp. 273–280 (cit. on p. 13).

[16] Wen-Xiang Fang, Po-Chao Lan, Wan-Rung Lin, Hsiao-Chen Chang, Hai-Yen Chang, and Yi-Hsien Wang. «Combine facebook prophet and LSTM with BPNN forecasting financial markets: the morgan Taiwan index». In: *2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*. IEEE. 2019, pp. 1–2 (cit. on p. 13).

[17] Kathleen Haggerty, Chao M Yuan, Benedict O Okoh, and Peter L Williamson. *Method and apparatus for development and use of a credit score based on spend capacity*. US Patent 7,890,420. Feb. 2011 (cit. on p. 13).

[18] Kathleen Haggerty, Chao Yuan, Benedict Okoh, and Peter Williamson. *Method and apparatus for targeting best customers based on spend capacity*. US Patent App. 11/169,778. Oct. 2006 (cit. on p. 13).

[19] Abdul Aziz and Gerald H Lawson. «Cash flow reporting and financial distress models: Testing of hypotheses». In: *Financial Management* (1989), pp. 55–63 (cit. on p. 23).

[20] Andrew C Harvey and Simon Peters. «Estimation procedures for structural time series models». In: *Journal of forecasting* 9.2 (1990), pp. 89–108 (cit. on p. 29).

[21] Sara Alaee, Kaveh Kamgar, and Eamonn Keogh. «Matrix profile XXII: exact discovery of time series motifs under DTW». In: *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2020, pp. 900–905 (cit. on p. 31).

[22] Bruce A Dautrich, Lawrence R Rabiner, and Thomas B Martin. «The Effects of Selected Signal Processing Techniques on the Performance of a Filter-Bank-Based Isolated Word Recognizer». In: *Bell System Technical Journal* 62.5 (1983), pp. 1311–1336 (cit. on p. 32).

[23] Peter Markstein. *Computational Systems Bioinformatics: CSB 2008 Conference Proceedings, Volume 7: Stanford University, USA, 26-29 August 2008*. Imperial College Press, 2008 (cit. on p. 32).

[24] Eamonn Keogh and Chotirat Ann Ratanamahatana. «Exact indexing of dynamic time warping». In: *Knowledge and information systems* 7.3 (2005), pp. 358–386 (cit. on p. 33).

[25] Toni M Rath and R Manmatha. «Lower-bounding of dynamic time warping distances for multivariate time series». In: *University of Massachusetts Amherst Technical Report MM* 40 (2002), pp. 1–4 (cit. on p. 34).

[26] Yue Lu, Renjie Wu, Abdullah Mueen, Maria A Zuluaga, and Eamonn Keogh. «Matrix Profile XXIV: Scaling Time Series Anomaly Detection to Trillions of Datapoints and Ultra-fast Arriving Data Streams». In: () (cit. on p. 33).

[27] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. «Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets». In: *2016 IEEE 16th international conference on data mining (ICDM)*. Ieee. 2016, pp. 1317–1322 (cit. on p. 33).

[28] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. «The M4 Competition: 100,000 time series and 61 forecasting methods». In: *International Journal of Forecasting* 36.1 (2020), pp. 54–74 (cit. on p. 38).

[29] Robert K Lai, Chin-Yuan Fan, Wei-Hsiu Huang, and Pei-Chann Chang. «Evolving and clustering fuzzy decision tree for financial time series data forecasting». In: *Expert Systems with Applications* 36.2 (2009), pp. 3761–3773 (cit. on p. 38).

[30] Donghwan Kim and Jun-Geol Baek. «Bagging ensemble-based novel data generation method for univariate time series forecasting». In: *Expert Systems with Applications* 203 (2022), p. 117366 (cit. on p. 38).

[31] Matthew Ward, Kevin Malmsten, Hassan Salamy, and Cheol-Hong Min. «Data Balanced Bagging Ensemble of Convolutional-LSTM Neural Networks for Time Series Data Classification with an Imbalanced Dataset». In: *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2021, pp. 1–5 (cit. on p. 38).

[32] Michael J Kane, Natalie Price, Matthew Scotch, and Peter Rabinowitz. «Comparison of ARIMA and Random Forest time series models for prediction of avian influenza H5N1 outbreaks». In: *BMC bioinformatics* 15.1 (2014), pp. 1–9 (cit. on p. 38).

[33] Zahra Putri Agusta et al. «Modified balanced random forest for improving imbalanced data prediction». In: *International Journal of Advances in Intelligent Informatics* 5.1 (2019), pp. 58–65 (cit. on p. 38).

[34] Héctor Allende and Carlos Valle. «Ensemble methods for time series forecasting». In: *Claudio moraga: A passion for multi-valued logic and soft computing*. Springer, 2017, pp. 217–232 (cit. on p. 38).

[35] Chris Seiffert, Taghi M Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. «RUSBoost: A hybrid approach to alleviating class imbalance». In: *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 40.1 (2009), pp. 185–197 (cit. on p. 38).

[36] John Paparrizos and Luis Gravano. «k-shape: Efficient and accurate clustering of time series». In: *Proceedings of the 2015 ACM SIGMOD international conference on management of data*. 2015, pp. 1855–1870 (cit. on p. 38).

[37] John Paparrizos and Luis Gravano. «Fast and accurate time-series clustering». In: *ACM Transactions on Database Systems (TODS)* 42.2 (2017), pp. 1–49 (cit. on pp. 38, 39).

[38] Alessandro Camerra, Themis Palpanas, Jin Shieh, and Eamonn Keogh. «isax 2.0: Indexing and mining one billion time series». In: *2010 IEEE International Conference on Data Mining*. IEEE. 2010, pp. 58–67 (cit. on p. 40).

[39] Christopher X Ren, Claudia Hulbert, Paul A Johnson, and Bertrand Rouet-Leduc. «Machine learning and fault rupture: a review». In: *Advances in Geophysics* 61 (2020), pp. 57–107 (cit. on p. 42).

[40] *scikit-learn*. `https://scikit-learn.org/stable/`. Accessed: 2021-12-23 (cit. on p. 52).

[41] Trevor Hastie, Saharon Rosset, Ji Zhu, and Hui Zou. «Multi-class adaboost». In: *Statistics and its Interface* 2.3 (2009), pp. 349–360 (cit. on p. 52).

[42] *R's autoarima documentation*. `https://www.rdocumentation.org/packages/forecast/versions/8.16`. Accessed: 2021-12-23 (cit. on p. 56).

[43] Michael Mainelli and Mark Yeandle. «Best execution compliance: new techniques for managing compliance risk». In: *The Journal of Risk Finance* (2006) (cit. on p. 62).

[44] Tatjana Pavlenko, Oleksandr Chernyak, and Annika Tillander. «Bayesian Networks for Modeling and Assessment of Credit Concentration Risks». In: *International Statistical Conference Prague. Available online: http://www. czso. cz/conference2009/proceedings/data/methods/pavlenko_paper. pdf (accessed on 21 July 2018)*. 2009 (cit. on p. 62).

[45] I-Cheng Yeh and Che-hui Lien. «The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients». In: *Expert Systems with Applications* 36.2 (2009), pp. 2473–2480 (cit. on p. 62).

[46] Tony Bellotti and Jonathan Crook. «Support vector machines for credit scoring and discovery of significant features». In: *Expert systems with applications* 36.2 (2009), pp. 3302–3308 (cit. on p. 62).

[47] Jorge Galindo and Pablo Tamayo. «Credit risk assessment using statistical and machine learning: basic methodology and risk modeling applications». In: *Computational Economics* 15.1 (2000), pp. 107–143 (cit. on p. 62).

[48] Hong Wang, Qingsong Xu, and Lifeng Zhou. «Large unbalanced credit scoring using lasso-logistic regression ensemble». In: *PloS one* 10.2 (2015), e0117844 (cit. on p. 62).

[49] Shigeyuki Hamori, Minami Kawai, Takahiro Kume, Yuji Murakami, and Chikara Watanabe. «Ensemble learning or deep learning? Application to default risk analysis». In: *Journal of Risk and Financial Management* 11.1 (2018), p. 12 (cit. on p. 62).

[50] Terry Harris. «Quantitative credit risk assessment using support vector machines: Broad versus Narrow default definitions». In: *Expert Systems with Applications* 40.11 (2013), pp. 4404–4413 (cit. on p. 62).

[51] Cheng-Lung Huang, Mu-Chen Chen, and Chieh-Jen Wang. «Credit scoring with a data mining approach based on support vector machines». In: *Expert systems with applications* 33.4 (2007), pp. 847–856 (cit. on pp. 62–65).

[52] Abbas Keramati and Niloofar Yousefi. «A proposed classification of data mining techniques in credit scoring». In: *the Proceeding of 2011 International Conference of Industrial Engineering and Operations Management, Kuala Lumpur, Malaysia, Jurnal*. 2011, pp. 22–4 (cit. on p. 63).

[53] Amir E Khandani, Adlar J Kim, and Andrew W Lo. «Consumer credit-risk models via machine-learning algorithms». In: *Journal of Banking & Finance* 34.11 (2010), pp. 2767–2787 (cit. on p. 63).

[54] Kin Keung Lai, Lean Yu, Ligang Zhou, and Shouyang Wang. «Credit risk evaluation with least square support vector machine». In: *International Conference on Rough Sets and Knowledge Technology*. Springer. 2006, pp. 490–495 (cit. on p. 63).

[55] Stefan Lessmann, Bart Baesens, Hsin-Vonn Seow, and Lyn C Thomas. «Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research». In: *European Journal of Operational Research* 247.1 (2015), pp. 124–136 (cit. on p. 63).

[56] Van-Sang Ha and Ha-Nam Nguyen. «Credit scoring with a feature selection approach based deep learning». In: *MATEC Web of Conferences*. Vol. 54. EDP Sciences. 2016, p. 05004 (cit. on p. 63).

[57] João A Bastos. «Ensemble predictions of recovery rates». In: *Journal of Financial Services Research* 46.2 (2014), pp. 177–193 (cit. on p. 64).

[58] Flavio Barboza, Herbert Kimura, and Edward Altman. «Machine learning models and bankruptcy prediction». In: *Expert Systems with Applications* 83 (2017), pp. 405–417 (cit. on p. 64).

[59] Reza Raei, Mahdi Saeidi Kousha, Saeid Fallahpour, and Mohammad Fadaeinejad. «A hybrid model for estimating the probability of default of corporate customers». In: *Iranian Journal of Management Studies* 9.3 (2016), pp. 651–673 (cit. on p. 64).

[60] Zijiang Yang, Wenjie You, and Guoli Ji. «Using partial least squares and support vector machines for bankruptcy prediction». In: *Expert Systems with Applications* 38.7 (2011), pp. 8336–8342 (cit. on p. 64).

[61] Maher Ala'raj and Maysam F Abbod. «Classifiers consensus system approach for credit scoring». In: *Knowledge-Based Systems* 104 (2016), pp. 89–105 (cit. on p. 64).

[62] Maher Ala'raj and Maysam F Abbod. «A new hybrid ensemble credit scoring model based on classifiers consensus system approach». In: *Expert Systems with Applications* 64 (2016), pp. 36–55 (cit. on p. 64).

[63] Dinesh Bacham and Janet Zhao. «Machine learning: challenges, lessons, and opportunities in credit risk modeling». In: *Moody's Analytics Risk Perspectives* 9 (2017), pp. 30–35 (cit. on p. 64).

[64] Jie Cao, Hongke Lu, Weiwei Wang, and Jian Wang. «A loan default discrimination model using cost-sensitive support vector machine improved by PSO». In: *Information Technology and Management* 14.3 (2013), pp. 193–204 (cit. on p. 65).

[65] Ir Tony Van Gestel, Bart Baesens, Ir Joao Garcia, and Peter Van Dijcke. «A support vector machine approach to credit scoring». In: *FORUM FINANCIER-REVUE BANCAIRE ET FINANCIAIRE BANK EN FINANCIEWEZEN* . Citeseer. 2003, pp. 73–82 (cit. on p. 65).

[66] Peter Martey Addo, Dominique Guegan, and Bertrand Hassani. «Credit risk analysis using machine and deep learning models». In: *Risks* 6.2 (2018), p. 38 (cit. on p. 65).

[67] Rashmi Malhotra and Davinder K Malhotra. «Evaluating consumer loans using neural networks». In: *Omega* 31.2 (2003), pp. 83–96 (cit. on p. 65).

[68] Aleksandra Wójcicka et al. «Neural networks vs discriminant analysis in the assessment of default». In: *Annales Universitatis Mariae Curie-Skłodowska, Sectio H Oeconomia* 51.5 (2017), pp. 339–349 (cit. on p. 65).

[69] Wenhao Zhang et al. «Machine learning approaches to predicting company bankruptcy». In: *Journal of Financial Risk Management* 6.04 (2017), p. 364 (cit. on p. 65).

[70] Tineke Blom. «Top down Stress Testing: An Application of Adaptive Lasso to Forecasting Credit Loss Rates». MA thesis. 2015 (cit. on p. 65).

[71] Mr Jorge A Chan-Lau. *Lasso regressions and forecasting models in applied stress testing*. International Monetary Fund, 2017 (cit. on p. 65).

[72] Michael Jacobs Jr. «The validation of machine-learning models for the stress testing of credit risk». In: *Journal of Risk Management in Financial Institutions* 11.3 (2018), pp. 218–243 (cit. on p. 65).

[73] Tushith Islam, Christos Vasilopoulos, and Erik Pruyt. «Stress-testing banks under deep uncertainty». In: *Proceedings of the 31st International Conference of the System Dynamics Society, Cambridge, Massachusetts, USA, 21-25 July 2013*. The System Dynamic Society. 2013 (cit. on p. 65).

[74] Jun-ya Gotoh, Akiko Takeda, and Rei Yamamoto. «Interaction between financial risk measures and machine learning methods». In: *Computational Management Science* 11.4 (2014), pp. 365–402 (cit. on p. 66).

[75] Jordi Petchamé Sala. «Liquidity risk modeling using artificial neural network». MA thesis. Universitat Politècnica de Catalunya, 2011 (cit. on p. 66).

[76] Madjid Tavana, Amir-Reza Abtahi, Debora Di Caprio, and Maryam Poortarigh. «An Artificial Neural Network and Bayesian Network model for liquidity risk assessment in banking». In: *Neurocomputing* 275 (2018), pp. 2525–2554 (cit. on p. 66).

[77] Iain Brown and Christophe Mues. «An experimental comparison of classification algorithms for imbalanced credit scoring data sets». In: *Expert Systems with Applications* 39.3 (2012), pp. 3446–3453 (cit. on p. 66).

[78] Heng-Guo Zhang, Chi-Wei Su, Yan Song, Shuqi Qiu, Ran Xiao, and Fei Su. «Calculating Value-at-Risk for high-dimensional time series using a nonlinear random mapping model». In: *Economic Modelling* 67 (2017), pp. 355–367 (cit. on p. 66).

[79] Babak Mahdavi-Damghani and Stephen Roberts. «A proposed risk modeling shift from the approach of stochastic differential equation towards machine learning clustering: Illustration with the concepts of anticipative & responsible VaR». In: *Available at SSRN 3039179* (2017) (cit. on p. 66).

[80] Soheil Almasi Monfared and David Enke. «Volatility forecasting using a hybrid GJR-GARCH neural network model». In: *Procedia Computer Science* 36 (2014), pp. 246–253 (cit. on p. 66).

[81] Loris Foresti, Devis Tuia, Vadim Timonin, and Mikhail Kanevski. «Time series input selection using multiple kernel learning». In: *Proceedings of the 18th European Symposium on Artificial Neural Networks-Computational Intelligence and Machine Learning, ESANN 2010*. 2010, pp. 123–128 (cit. on p. 66).

[82] Gareth Peters, Pavel V Shevchenko, Ruben Cohen, and Diane Maurice. «Statistical machine learning analysis of cyber risk data: event case studies». In: *Available at SSRN 3073704* (2017) (cit. on p. 66).

[83] Joseph Pun and Yuri Lawryshyn. «Improving credit card fraud detection using a meta-classification strategy». In: *International Journal of Computer Applications* 56.10 (2012) (cit. on p. 67).

[84] Shashank Sharma and Arjun Roy Choudhury. «Fraud analytics: A survey on bank fraud and fraud prediction using unsupervised learning based approach». In: *International Journal of Innovations in Engineering Research and Technology* 3.3 (2016), pp. 1–9 (cit. on p. 67).

[85] Eric WT Ngai, Yong Hu, Yiu Hing Wong, Yijun Chen, and Xin Sun. «The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature». In: *Decision support systems* 50.3 (2011), pp. 559–569 (cit. on p. 67).

[86] Agus Sudjianto, Sheela Nair, Ming Yuan, Aijun Zhang, Daniel Kern, and Fernando Cela-Dıaz. «Statistical methods for fighting financial crimes». In: *Technometrics* 52.1 (2010), pp. 5–19 (cit. on p. 67).

[87] Marina Pavlovna Khrestina, Dmitry Ivanovich Dorofeev, Polina Andreevna Kachurina, Timur Rinatovich Usubaliev, and Aleksey Sergeevich Dobrotvorskiy. «Development of algorithms for searching, analyzing and detecting fraudulent activities in the financial sphere». In: (2017) (cit. on p. 67).