Universität Zürich

Machine Learning for Economic Analysis

Machine Learning for Credit Card Fraud Detection

University of Zurich[UZH]

Andrianos Michail, Hatem Khrouf, Jessica Rey, Matthias Leuthard and Nina Reiser

Supervisors: Professor Damian Kozbur, Matteo Courthoud

March 9, 2021

Contents

List of Figures

List of Tables

1 Introduction

1.1 Credit Card Fraud as an Economic Issue

Fraudulent credit card transaction are the source of billions of dollars of loss every year (Dal Pozzolo et al., 2014; Dal Pozzolo, 2015). The European Central Bank (ECB) oversight report on card fraud analysis estimates the total value of card fraud using cards issued in Single Euro Payments Aera (SEPA) to €1.8 billion, while the total value of card transactions using cards issued in in SEPA is estimated at €4.38 trillion in 2016 (ECB, 2018). It must additionally be noted, credit-card fraud estimations realistically only measure the loss of frauds that have been detected or declared by the customer.

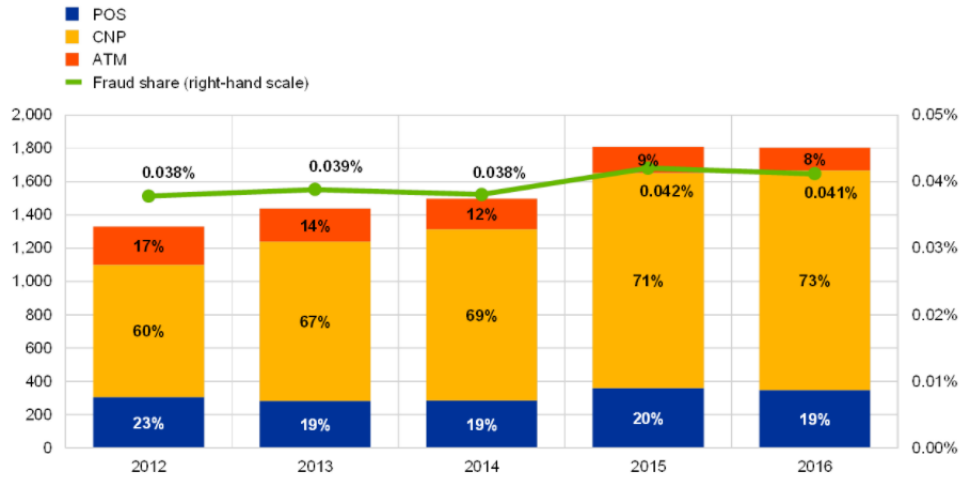

Figure 1: Evolution of the total value of card fraud using cards issued within SEPA; left-hand scale: total value (EUR millions); right-hand scale: value of fraud as share of value of transaction

Source: https://www.ecb.europa.eu/pub/cardfraud/html/ecb.cardfraudreport201809.en.html

In the years leading to the report, and as exhibited in Figure 1, card fraud has been increasing in importance. Card Not Present (CNP) fraud is of growing concern, and is a specific type of credit card fraud in which the fraudulent transaction is done without physically having the card (ECB, 2018).

Credit card fraud affects all types of people; commercial entities, banks, but also private citizens (Dal Pozzolo, 2015). The cost of fraud is two fold; if a bank loses money due to fraud, the cost will eventually be borne by clients through higher interest rates or membership fees (Dal Pozzolo, 2015), not to mention the potentially unrecovered losses a person may suffer from a fraudulent transaction. It may also be costly to a financial institution through its reputation; though unquantifiable, if cardholders are victims of fraud, they may not view that company as trustworthy (Dal Pozzolo, 2015).

The prevention of fraud would make card payment more attractive to businesses and customers (Snellman et al., 2001), as better fraud prevention is associated with a higher reliance on card payments instead of cash (Hoffmann and Birnbrich, 2012). This is a topic of interest to economists and policy makers alike, since increasing the use of cards improves economic welfare of both consumers and businesses (Garcia-Swartz et al., 2006), contributes to tackling shadow economies (Schneider, 2013) and is more cost-effective than using cash (Bergman et al., 2007).

Finally, there is some ethical relevance to the detection of fraudulent credit card transaction. As identified by Everett (2009), credit card fraud is an issue broader than simple petty crime: it helps fund organised crime, international narcotics trafficking, and even terrorist financing.

1.2 Aims and Objectives

To successfully take action against credit card fraud, one must not only prevent the fraud (by blocking it with necessary means) but also, and perhaps more importantly, detect it. The process of fraud detetion is correctly identifying if a transaction belongs to a class of fraudulent or genuine transactions (Dal Pozzolo, 2015).

Considering the volume of financial transactions, automatic systems through algorithms (as opposed to human screening) are key to detection of fraudulence and reducing these losses (Andrea Dal Pozzolo et al., 2014). Predictive models are already in active use today, yet, data mining approaches for fraud are seldom used (Siddhartha Bhattacharyya, 2011). With this project, we use Machine Learning techniques to train an algorithm to detect credit card fraud. If Machine Learning is promising in this regard, fraud could be further prevented upon transaction which would result in several aforementioned benefits.

Businesses are also favourable to refined detection techniques, over other authentication methods (such as 3-D secure) from fear they may cause poor customer experience and frustration leading to a risk of a customer abandoning their purchase (Dal Pozzolo, 2015).

1.3 Challenges

Designing efficient algorithms to detect fraud is challenging, particularly due to the continuous stream of transactions and highly imbalanced data (Andrea Dal Pozzolo et al., 2014). The first issue is addressed through the structure of our data set: it is a finite set of transactions over a period of sampling of two days. The second issue, imbalanced data, is addressed throughout the project. In Section 2 the issue of imbalanced data is described, and in Section 3, three data balancing methods used in practice in this paper are presented.

From a learning perspective, the design of fraud detection algorithms is according to Dal Pozzolo et al. (2014); Dal Pozzolo (2015); Dal Pozzolo et al. (2017) particularly challenging, since fraud detection is characterized by concept drift (changes in customer behavior and fraudsters' ability to invent novel fraud patterns), verification latency (only a small set of transactions are timely checked by investigators) and class imbalance (Dal Pozzolo et al., 2014; Dal Pozzolo, 2015; Dal Pozzolo et al., 2017).

1.4 Overview of the Report

In this body of work, we will start by analyzing our data and its shape in Section 2. Thereafter, Section 3 discusses the methods used to transform our data (min/max scaling) and provides an overview of the Data Balancing Techniques (Over- and Undersampling, Autoencoders) we applied in this project. It continues by presenting the main classification algorithm (Logistic Regression) and by describing the evaluation metrics (Precision, Recall, F1) to give insight in our choices, and detail the experimental steps applied. In Section 4 results are presented. Finally, conclusion is drawn Section 5 and limitations encountered within the project are discussed.

2 Data Shape and Analysis

In this analysis,the Credit Card Fraud Detection data set Kaggle (2017) is used that contains transactions carried out by credit cards in September 2013 by European cardholders. This data set shows transactions that occurred in two days and was created by the Machine Learning Group at Universite Libre de Bruxelles. It is open source and available on Kaggle (Kaggle, 2017).

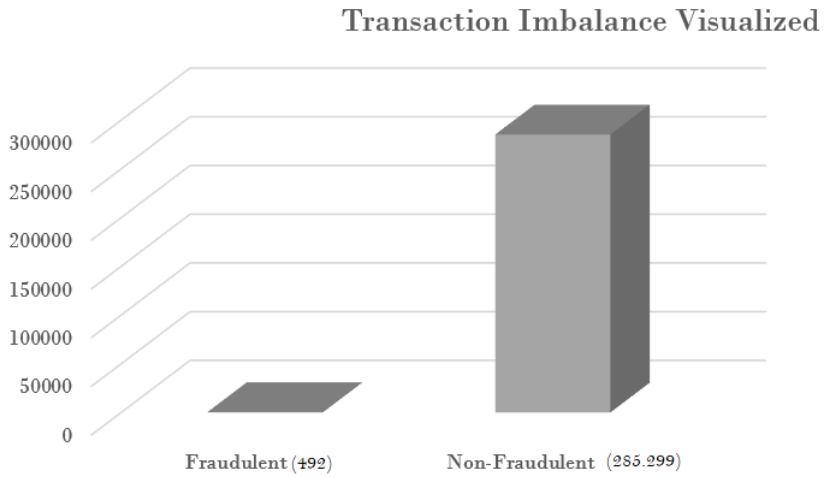

Figure 2: Transaction Imbalance

The outcome variable in credit card fraud data sets is usually extremely imbalanced where the number of fraudulent transactions is much lower than the number of instances classified as non-fraudulent transactions (Dal Pozzolo et al., 2014; Krivko, 2010). More specifically, frauds contribute typically less than 1 percent to the overall transactions (Krivko, 2010; Dal Pozzolo et al., 2014). Likewise, our dataset includes only 492 fraudulent transactions out of 285.299 which corresponds to a very low fraud proportion of only 0.172 percent. Therefore, the incidence of credit card fraud is limited to a small percentage of transactions which is visualized in Figure 2. According to Li and Sun (2012) and Mujalli et al. (2016), a dataset is considered to be imbalanced if the proportion of the minority class sample constitutes less than 35 percent of the dataset. Therefore, we face the challenge of highly imbalanced class distribution in our dataset since fraudulent transactions, which are of more interest since they cause huge financial losses, are underrepresented relative to non-fraudulent transactions.

The data set contains 28 variables that are only numerical, a result of Principal Component Analysis (PCA) transformation (see Section 2.2). Feature "Time" contains the seconds elapsed between each transaction and feature "Amount" is the transaction amount. Finally, "Class" is the dependent binary variable which takes on values of 1 in case of fraud, and 0 otherwise. Its frequency distribution is exemplified in Figure 2.

2.1 Class Distribution Discussion

Imbalanced class distribution of the outcome variable is a crucial problem in ML (Ali et al., 2015; Ertekin et al., 2007; He and Garcia, 2009; Sun et al., 2009) and its handling is a controversially discussed issue (Nguyen et al., 2009; Krawczyk, 2016; Lemaître et al., 2017). Imbalanced datasets are common in

several domains, including but not limited to: fraud detection, ad serving, traffic accidents and medical diagnostic/disease screening Lemaître et al. (2017). The problem is specifically common in classification problems (Lemaître et al., 2017) and is associated with a poor predictive performance of most standard classifier learning algorithms (Nguyen et al., 2009; Krawczyk, 2016). The classifiers tend to be biased towards the majority class (within this task non-fraudulent credit cards) and fraudulent credit cards are more likely to be miss-classified (Nguyen et al., 2009; He and Garcia, 2009; Krawczyk, 2016). This is because of the available information: it is more challenging for models to learn the defining characteristics of the minority class, whose prediction is often that of interest. However, within this task the correct classification of fraudulent credit cards is from a learning point of view of great value due to its high monetary miss-classification costs.

Within this report the problem of the imbalanced outcome variable is addressed by data balancing techniques at both the data-level (Over and Undersampling, Section 3.2) and using a form of semi-supervised learning using Autoencoders at section 3.4.

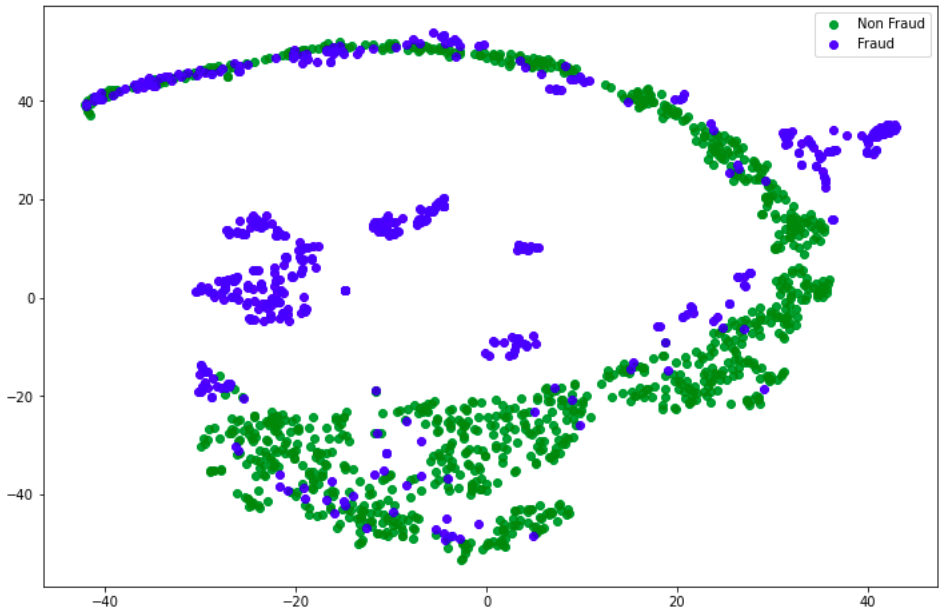

Figure 3: 2D (TtSNE) Visualization of input data(PCA)

Figure 3 is a 2D representation of the data acquired using t-Distributed Stochastic Neighbor Embedding (T-SNE) which produces a graph of reduced dimensions for visualisation purposes (Maaten and Hinton, 2008). Many fraud and non fraud samples seem to have similar features, thus are harder to separate with a classifier without further preprocessing.

2.2 Principal Component Analysis

For confidentialty reasons, the original features of the dataset can not be provided and the data has been encrypted using PCA. Only the two features "Time" and "Amount" have not been transformed with PCA. The dataset contains exclusively numerical values, which are the result of a PCA transformation.

To interpret high dimensional datasets a reduction method which preserves important information in the data is required (Jolliffe and Cadima, 2016). PCA is a multivariate technique to analyze observations described by several dependent variables which often are inter-correlated (Abdi and Williams, 2010; Hastie et al., 2009). The method reduces the dimensionality of large datasets by transformation without losing important information or variability (Jolliffe and Cadima, 2016). Data preprocessing is another application of PCA. Therefore, data with various measurements or different variability is standardized in order apply recognition algorithms successfully (Dreiseitl and Ohno-Machado, 2002). PCA improves the performance of ML methods and therefore the classification performance (Erkmen and Yildirim, 2008; Howley et al., 2005).

PCA identifies patterns, similarities and differences in the data and computes new variables that are linear combinations of the original variables. The new variables or principal components maximize the variable variance and are uncorrelated with each other (Abdi and Williams, 2010; Jolliffe and Cadima, 2016). The first principal component is calculated under the requirement of having the largest variance. The second principal component must be orthogonal to the first principal component and therefore uncorrelated. The second principal component additionally requires to have the second high variance in the dataset. The following principal components are computed under the same requirements (Abdi and Williams, 2010).

2.3 Data Split

From the 285,299 available observations, a random 90 percent of observations were used to train the data. The remaining 10 percent were used to test it (30'000 entries). The split of the data was then rebalanced, as is discussed in Section 3.

3 Methodology and Experiments

In this section, different Data-Scaling methods and data balancing techniques are presented. In general, Data Scaling is a fundamental part of preprocessing data. In this project we apply min-max scaling in order to optimize model performance. Further, to address the problem of highly imbalanced data, we employ data balancing techniques such as Under- and Oversampling and SMOTE which can help to improve the prediction accuracy of the classifiers. Under- and Oversampling are data-level techniques whereas SMOTE is just an Oversampling approach (Chawla et al., 2002). To improve performance further, we additionally perform experiments based on the Autoencoders technique in order to extract hidden representations. The classification algorithm Logistic Regression then estimates the class membership probability of the observations (James et al., 2017). Evaluation metrics such as Precision, Recall and the F1-measure are used to evaluate the prediction performance.

3.1 Preprocessing

3.1.1 Feature Scaling - Min/Max Scaling

Feature scaling is a data preprocessing method used to normalize the range of features. Therefore, it corrects the overestimated weight of some features relative to others which otherwise could lead to erroneous predictions (Forman, 2006).

For this project, min-max scaling is used as feature scaling method which rescales the range of features to [0,1] (Zheng and Casari, 2018). The general formula for a min-max of [0,1] is:

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

Where x is the original value and x' is the normalized value.

To rescale a range between an arbitrary set of values [a,b], the formula becomes:

$$x' = a + \frac{(x - min(x))(b - a)}{max(x) - min(x)} \tag{2}$$

3.2 Data Balancing Techniques: Oversampling and Undersampling

The predictive accuracy is important to evaluate the performance of the Machine Learning method (Chawla et al., 2002). At times, collection of more data to balance the classes is unavailable. To overcome the problem of poor prediction accuracy of highly imbalanced data, the original dataset is resampled by the following methods. The dataset is resampled by Oversampling the minority class, by Undersampling the majority class or both (Chawla et al., 2002).

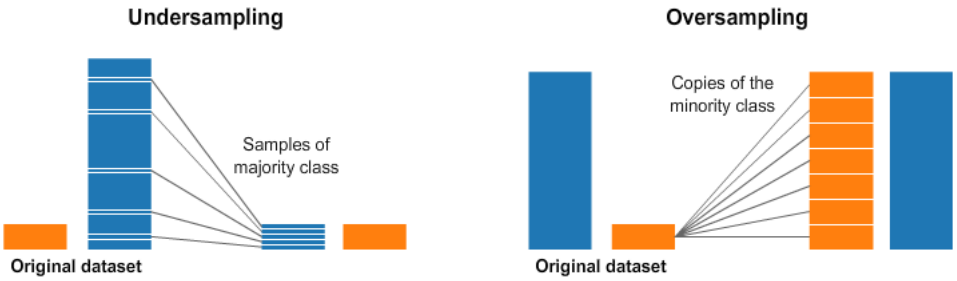

Figure 4: Undersampling and Oversampling

Source: https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasetst1

Resampling methods are data-level techniques to rebalance highly unbalanced data sets. As schematized in Figure 4, two resampling methods are at our disposal: Undersampling and Oversampling. Undersampling is the act of removing random observations from the majority class until it equals the size of the minority class (Japkowicz et al., 2000). Down-sizing the majority or negative class could cause a loss of information (Japkowicz et al., 2000). Oversampling is the method of duplicating random observations from the positive or minority class until it matches the size of the other class (Japkowicz et al., 2000). The approach balances the class distribution without adding new information to the data set, which might lead to overfitting the data (García et al., 2012; Lemaître et al., 2017). The study of Japkowicz et al. (2000) imply that both methods are very effective to improve the accuracy of a classifier on imbalanced datasets. But different studies imply that on large datasets, Undersampling often results in better prediction accuracy than Oversampling (García et al., 2012; Japkowicz et al., 2000).

Another approach is a hybrid method, that combines different data balancing techniques. The advantage of this method is that the bias of Under- and Oversampling are from a different kind of nature (Estabrooks et al., 2004).

3.3 Data Balancing Technique: Synthetic Minority Oversampling Technique

In Undersampling, we lose information of the majority data set. Synthetic Minority Oversampling Technique (SMOTE) is an Oversampling approach proposed by Chawla et al. (2002) in which the minority class is over-sampled through the synthetic creation of minority samples, rather than by Oversampling with replacement, as is the case in 3.2. This process is done by analysing the feature space of the minority samples and identifying for each observation their k nearest minority-class neighbours. The synthetic samples are generated by taking the difference between the feature vector and its nearest neighbour, and then multiplying the difference by a random number within a $[0, 1]$ range and adding the result to the feature vector under consideration (Chawla et al., 2002), as schematized in 5. The result is a random point along the segment between two features.

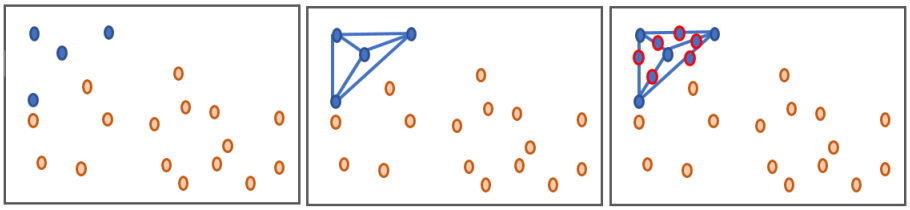

Figure 5: SMOTE Algorithm, Schematized

Source: https://datasciencecampus.github.io/balancing-data-with-smote/

Parameters of the algorithm are the number of minority class samples, the amount of Oversampling (SMOTE in percent) and number of nearest neighbours k.

In their paper Chawla et al. (2002) show that SMOTE improves the accuracy of classifiers for a minority class. This improvement, in comparison to Oversampling with replacement, can be attributed to the larger decision region of the minority class created through SMOTE.

3.4 Extraction of Hidden Representations: Autoencoders

This chapter about the Autoencoder technique is based on Sweers et al. (2018); Al-Shabi (2019); Misra et al. (2020) and Kingma and Welling (2019). Autoencoder is an Artificial Neural Network used to learn efficient data codings in an Unsupervised (or sometimes Semi Supervised) manner. The Autoencoder aims to learn a representation (encoding) for a set of data, typically used for dimensionality reduction, so the resulting encoding usually has a smaller dimension than the input data. The Autoencoder's Neural Network structure learns to map from input to output through a pair of encoding and decoding phases so that the input and the reconstruction of the encoding are close to each other. In its simplest form, the Autoencoder's Neural Network structure consists of three layers which are graphically displayed in Figure 6. The transition between the first and the second layer represents the encoder and the decoder maps the hidden layers to the output layer to reconstruct the inputs. The first layer represents the input layer and exhibits m nodes which is the same amount of nodes as the dimension of the data. The middle hidden layer consists of n nodes, where n is the dimension of the encoding. The third layer is called the output layer and similar to the input layer has m nodes.

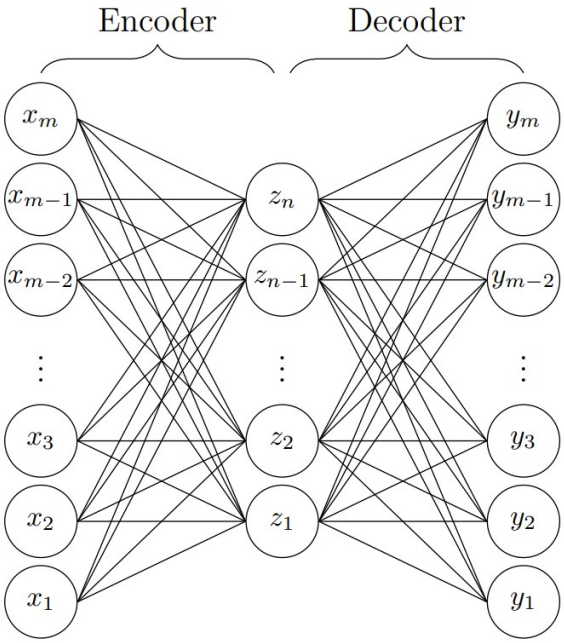

Figure 6: Basic Autoencoder Diagram

Source: Sweers et al. (2018)

Based on a Backpropagation algorithm the method applies the encoding and decoding procedure in order to minimize reconstruction error which is a metric measuring the similarity between input and output of the Autoencoder. Hereby, the mean squared error is frequently used as reconstruction error. During the training phase the Autoencoder algorithm is fed with unlabeled training data and it aims to minimize the reconstruction error over all training data.

To recognise fraud in the credit card data set we consider the above introduced Autoencoder technique but we apply a slightly more complex structure using two hidden layer for encoding and two hidden layer for decoding. In the literature (Sweers et al., 2018; Al-Shabi, 2019; Kingma and Welling, 2019), this type of Autoencoder is sometimes referred to as deep Autoencoder or stacked Autoencoder. In our setting, the Autoencoder method reduces the dimension of the data which we call the hidden representation. The reconstruction error captures the difference between non-fraudulent and fraudulent transactions in the hidden representation and therefore, can be considered as an anomaly score. More specifically, the procedure to detect fraud credit cards by using Autoencoders looks as follows: First, we train the Autoencoder on a training set that only contains the non-fraudulent transactions. Since the Autoencoder is fitted on a subset of the dataset without encountering any fraudulent transactions, the reconstruction should perform more consistently on the non fraud than the fraud cases, therefore generating more distinctive features to be fed in the Logistic Regression classifier.

3.5 Classification Algorithm: Logistic Regression

In this Section, Logistic Regression for binary (two classes) classification is introduced. The Logistic Regression is a classification algorithm, which calculates the class membership probability of the observations to a particular category (James et al., 2017; Dreiseitl and Ohno-Machado, 2002). The output y

is categorical and denotes the two classes as 0 and 1 (Bishop, 2006). Probabilities range between 0 and 1 along the Sigmoid curve and are classified as either fraudulent or non-fraudulent based on a threshold. Any observation with a probability larger than this threshold will be classified as fraudulent, and non-fraudulent otherwise based on the probability of their belonging to either category.

The classification algorithm Logistic Regression predicts the class membership probabilities of observations (Dreiseitl and Ohno-Machado, 2002; James et al., 2017). The components of vector x are called input variables and y denotes the class membership, which is binary (Dreiseitl and Ohno-Machado, 2002). The relationship between x and y is described by a probability distribution $p(x, y)$ and the optimal choice of class membership involves the maximization of the posterior distribution $p(y|x)$. The posterior probability of class 1 can be computed as a Logistic Sigmoid function,

$$p(1|x, w) = \sigma(w^T x) \tag{3}$$

and $p(0|x, w) = 1 - p(1|x, w)$. Modelling the given inputs variables x with parameters w calculates the posterior probabilities for both classes as $p(y|x) = f(x, w) \in [0, 1]$. The posterior distribution is dependent on the parameter w. It is based on the data-set and usually determined by the maximum-likelihood estimation (Dreiseitl and Ohno-Machado, 2002). The Sigmoid function for binary classification is

$$\sigma(z) = \frac{1}{1 + exp(-z)} \tag{4}$$

and the S-shape of the function is visualized in Figure 7 (Bishop, 2006). The function plots the real axis into a finite interval (James et al., 2017).

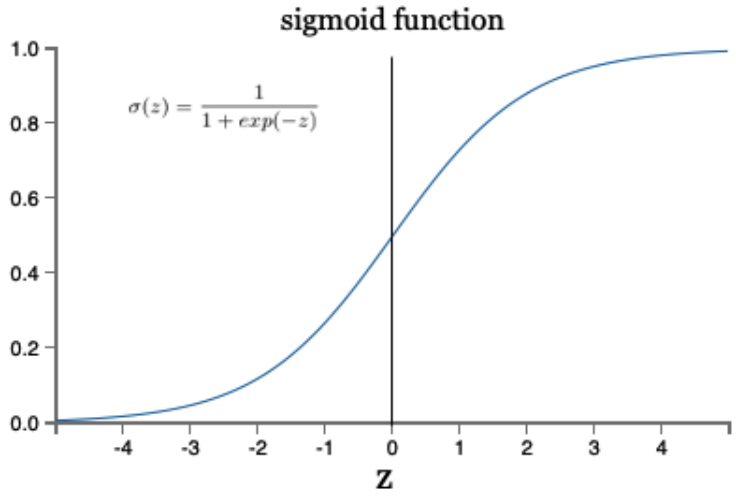

Figure 7: Sigmoid function

http://neuralnetworksanddeeplearning.com/chap1.html

The Discriminative classification method maps each feature x directly onto a class using the Discriminant function $f(x, w) \in [0, 1]$ (Bishop, 2006). To estimate the optimal parameters w directly, the

maximum likelihood estimation is applied (Dreiseitl and Ohno-Machado, 2002). This requires the maximization of the following expression:

$$\prod_{i=1}^{n} p(y_i|x_i, w) \tag{5}$$

Logistic Regression can be interpreted as simple Feed Forward Neural Network (FFNN) with no hidden layers if the Sigmoid function is used as an activation function (Dreiseitl and Ohno-Machado, 2002). Both Logistic Regression and FFNN use a function f and a parameter vector w to posterior probabilities for classes y as $p(y|x) = f(x, w)$ (Dreiseitl and Ohno-Machado, 2002).

Within the experiments conducted, Logistic Regression outperformed FFNN and Random Forest (RFs) and was deemed the appropriate classification algorithm in the scope of our exploration and only the Logistic Regression results are presented.

3.6 Evaluation Metrics

The application of ML models in the context of imbalanced data tend to produce high predictive accuracy over the majority class, but poor predictive accuracy over the minority class (García et al., 2010; López et al., 2012; Nguyen et al., 2009). This is because accuracy only gives information on the classifier's overall ML performance. Consequently, accuracy captures the proportion of instances which are correctly classified among all instances and does not distinguish between the number of correct labels of different classes, which in the framework of imbalanced data may lead to erroneous conclusions (García et al., 2010; López et al., 2012; Nguyen et al., 2009).

Within this task, a naive ML algorithm that classifies all instances to non fraud achieves high classification accuracy close to 99.8 percent, which is equal to the proportion of the non-fraudulent class. This is not a desired classifier and therefore accuracy is not considered as an appropriate metric for the analysis. Instead, we use other metrics such as Precision, Recall and F1-measure in order to evaluate model performance (García et al., 2010; López et al., 2012; Nguyen et al., 2009).

3.6.1 Precision

Precision is a metric that quantifies the number of correct positive predictions. It is calculated using the ratio of correct predictions of this class out of all predictions which corresponds to the accuracy of the class (Sokolova and Lapalme, 2009).

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{6}$$

The results vary from 0.0, representing no Precision, to 1.0 for full Precision or equivalently perfect prediction for the respective class.

3.6.2 Recall

Recall differs from Precision and provides an indication of missed class predictions. The measure quantifies the number of correct class predictions made out of all potential class predictions (Sokolova and Lapalme, 2009).

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{7}$$

The results vary from 0.0 , representing no Recall, to 1.0 for full Recall (Sokolova and Lapalme, 2009).

3.7 F1-measure

The F1-measure is the harmonic mean of Precision and Recall and allows the maximization of both rather than maximizing one and sacrificing the other He (2013).

$$F1\text{-}measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{8}$$

Analogous to both Recall and Precision, the F-Measure ranges from 0.0 for a poor score, to 1.0 representing a perfect score.

The F1-macro measure is a harmonic mean of macro Precision and macro Recall with the macro Precision and Recall defined respectively as follows (Chinchor, 1992):

$$MacroPrecision = \frac{1}{N} \sum Precision \tag{9}$$

$$MacroRecall = \frac{1}{N} \sum Recall \tag{10}$$

The F1-macro measure is the preferred version of F1-measure within our setting as we are aiming to maximize the precision and recall of both classes in within the predictions. (Chinchor, 1992). The F1-macro measure is defined as:

$$F1\text{-}macro\ measure = \frac{2 * MacroPrecision * MacroRecall}{MacroPrecision + MacroRecall} \tag{11}$$

3.7.1 Precision vs. Recall, Economic perspective and F1-measure

Deciding whether to maximize Precision or Recall will have a different impact in economic terms. In fact, maximizing Precision will minimize the false positives, while maximizing Recall will minimize false negatives (Sokolova and Lapalme, 2009). Taking into account the problem at hand, maximizing for Precision translates into minimizing the detection of fraud that isn't actually fraud. Erroneously detecting fraud could be costly for private citizens, who would suddenly see their transactions and accounts blocked without good reasons.

Maximizing for Recall translates into minimizing the number of realized frauds we miss. This is of course extremely important, given the nature of the problem. The goal is to detect the most instances of fraud, and not let any fraudulent transactions through the system.

Considering the paragraph above, it is very important for a bank to maximize both measures jointly, to detect all instances of fraud discreetly. The issue in trying to maximize measure without hurting the other is difficult in practice. This is why we turn to F-Measure, which combines both metrics into a single score (He, 2013).

3.8 Experimental Setup

The first step of the experiments is to preprocess the features using Min/Max scaling. Following Min/Max scaling, the experiment consisted of applying several data balancing techniques. The following approaches were tested:

- Undersampling, with a 3:1 proportion of non-fraudulent transactions (1323) to fraudulent transactions (441).

- Undersampling of the majority class to match the minority class, both equal to 441.
- Hybrid of Oversampling and Undersampling for both majority class and minority class to be equal to 1000 samples.
- Hybrid of Oversampling and Undersampling for both majority class and minority class to be equal to 2000.
- SMOTE experiment, with 1 to 3 distributions, 1017 fraudulent samples to 3081 non-fraudulent samples.
- SMOTE experiment, with 1 to 10 distributions, 1017 fraudulent samples to 10170 non-fraudulent samples.

The various experiments aim to provide the best results for data classification to different classes. Undersampling is expected to improve prediction accuracy compared to unbalanced data but downsizing the domain is leading to a loss of information (Japkowicz et al., 2000). Combining Under- and Oversampling in a hybrid approach could increase accuracy due to the different nature of the prediction biases (Estabrooks et al., 2004). The data balancing technique SMOTE is expected to increase the accuracy of classifiers compared to Oversampling (Chawla et al., 2002). For each data balancing techniques various distributions are used. The classification algorithm Logistic Regression is applied to the rebalanced data and the results of these different experiments and their Logistic Regression application are summarized in Section 4.

Different instances of Autoencoders are applied on the same Over- and Undersampling variations of data and subsequently used to train the Logistic Regression. The results are summarized in Section 4.

4 Results

The probability of class membership is modelled with the Logistic Regression on the different experimental setups and results are presented in this Section. The macro Precision, macro Recall and macro F1-measure are elaborated for the plain experiment and for the set-up with Autoencoders applied on the data. Additionally, the Autoencoded data is visualised. The plain Logistic Regression macro average results are presented in Table 1.

Data Split	**Precision**	**Recall**	**F1-measure**
Undersampling, 3:1	0.91	0.83	0.86
Undersampling 441:441	0.93	0.67	0.75
Hybrid 1000:1000	0.95	0.61	0.67
Hybrid 2000:2000	0.95	0.58	0.63
1000 SMOTE, 1:3	0.93	0.72	0.79
1000 SMOTE, 1:10	0.91	0.84	0.87

Table 1: Plain Logistic Regression Macro Average Results

Over- and Undersampling for both the majority and minority class to be equal to 1000 (resp. 2000) provides with 0.95 the best Precision score. 95% correct positive predictions are made and only 5% are false allocated to the class fraudulent. The SMOTE experiment with 1017 fraudulent and 10170 non-fraudulent (1 to 10 distribution) training data conclude with 0.84 in the best Recall value. The importance of maximizing the Recall value is to detect the most instances of fraud. 84% of fraudulent transactions are detected but 16% are erroneously classified as non-fraudulent and consequently slipped through the system. The maximization of both measures is essential in fraud detection, thus reflected in the jointly F1-measure. The highest F1-measure is reached with the following two experiments. Undersampling with 1323 non-fraudulent transactions and 411 fraudulent transaction (3 to 1 distribution) results in 0.86 F1-measure and the rebalancing technique SMOTE with a 1 to 10 distribution provides the F1-measure of 0.87. Both Hybrid methods provide a good Precision but with relatively lower Recall value, thus performing worse in the F1-measure.

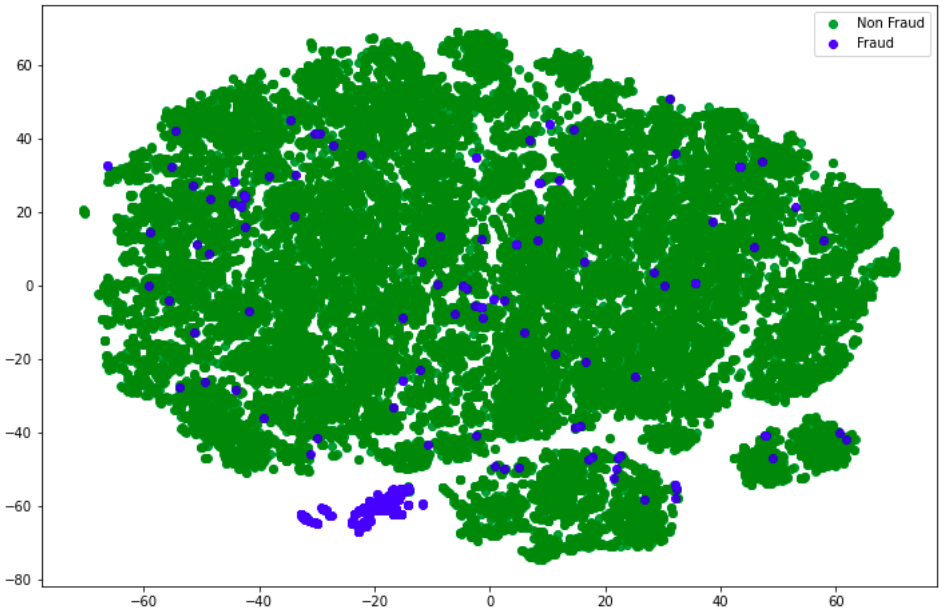

Figure 8: 2D Visualization of Hidden Representation Data

Similar to Figure 3, Figure 8 demonstrates a T-SNE visualisation of the test data after being processed through Autoencoders. As it can be seen, the Autoencoder has managed to distinguish the majority of fraud cases and provide an appropriate hidden representation, thus allowing the classifier to perform better.

Data Split	**AE Precision**	**AE Recall**	**AE F1-measure**
Undersampling, 3:1	0.89	0.93	0.91
Undersampling 441:441	0.91	0.84	0.87
Hybrid 1000:1000	0.93	0.79	0.83
Hybrid 2000:2000	0.94	0.72	0.79
1000 SMOTE, 1:3	0.91	0.90	0.89
1000 SMOTE, 1:10	0.88	0.93	0.90

Table 2: Macro Average Results with Autoencoders

Table 2 presents the result of the various experimental set-up with the Autoencoders applied on the data. Similar results are observed as in the plain experiment before regarding the different rebalancing techniques. The hybrid method provides a balance of good Precision and Recall, thus receiving a high F1-measure as well. Undersampling and SMOTE provide an improved F1-measure. Differentiating with the plain experiments, the Autoencoders improved the Precision performance of the classifier significantly. This proves that Autoencoder worked to improve classifying the data into the two classes fraudulent an non-fraudulent.

Finally, choosing the best result is a subjective exercise that depends on what the institution in question is trying to achieve. If, as suggested in previous sections of this paper, a bank is trying to operate a discrete process of fraud detection, and thus attempting to minimize falsely detecting fraud, the institu-

tion will favour a technique offering higher Precision score for equivalent F1-measures. Alternatively, if the institution is using this process to prevent fraud at all costs, it will favour a method offering a higher Recall for equivalent F1-measures.

5 Limitations and Conclusions

5.1 Limitations

Fraud detection problems are addressed in one of two ways: either in a static learning setting as is done in this paper, or in an online setting in which the detection model is updated instantly with data (Andrea Dal Pozzolo et al., 2014). In a credit card fraud detection problem such as the one tackled in this paper, we are of course limited to static learning.

Credit card fraud detection as a whole also suffers from the scarcity of available data (Andrea Dal Pozzolo et al., 2014), often bound by confidentiality agreements by issuers. Within this project, the data was given in a PCA format, disallowing experiments with the original data.

Within the scope of the project, the experiments aimed to find the best model and Autoencoder had been limited. It is expected that improved results could be achieved with further experiments with bigger variety of techniques hence the models introduced are not optimum.

5.2 Conclusions

Depending on the task and form of the data, Over- and Undersampling improves model performance. Moreover, Autoencoders as a form of Semi-Supervised Learning provides a better representation of the imbalanced data and therefore raises performance of the classification algorithms. Within the experiments carried out in this project, a Logistic Regression classification model performs better when utilising hidden representations as features. Further work using hidden representations to improve performance of other classification tasks for Economic purposes is advised. The high performance of this model is great news for the banks/businesses interested in maximizing security for their customers. This classifier is only one of the components necessary for fraud prevention but it is a crucial one.

References

Abdi, H. and Williams, L. J. (2010), 'Principal component analysis', *Wiley interdisciplinary reviews: computational statistics* **2**(4), 433–459.

Al-Shabi, M. (2019), 'Credit card fraud detection using autoencoder model in unbalanced datasets', *Journal of Advances in Mathematics and Computer Science* pp. 1–16.

Ali, A., Shamsuddin, S. M., Ralescu, A. L. et al. (2015), 'Classification with class imbalance problem: a review', *Int. J. Advance Soft Compu. Appl* **7**(3), 176–204.

Andrea Dal Pozzolo, O., Caelen, Y.-A., Le Borgne, S. and Waterschoot, G. B. (2014), 'Learned lessons in credit card fraud detection from a practitioner perspective', *Expert Systems with Applications* **41**, 4914–4928.

Bergman, M. A., Guibourg, G. and Segendorf, B. L. (2007), 'The Costs of Paying - Private and Social Costs of Cash and Card Payments', *Riksbank Research paper series* (212).

Bishop, C. M. (2006), *Pattern recognition and machine learning*, springer.

Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. (2002), 'Smote: Synthetic minority over-sampling technique', *Journal of Artificial Intelligence Research* **16**, 321–357.

Chinchor, N. (1992), 'Muc-4 evaluation metrics in proc. of the fourth message understanding conference 22–29'.

Dal Pozzolo, A. (2015), 'Adaptive machine learning for credit card fraud detection'.

Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C. and Bontempi, G. (2017), 'Credit card fraud detection: a realistic modeling and a novel learning strategy', *IEEE transactions on neural networks and learning systems* **29**(8), 3784–3797.

Dal Pozzolo, A., Johnson, R., Caelen, O., Waterschoot, S., Chawla, N. V. and Bontempi, G. (2014), Using hddt to avoid instances propagation in unbalanced and evolving data streams, *in* '2014 International Joint Conference on Neural Networks (IJCNN)', IEEE, pp. 588–594.

Dreiseitl, S. and Ohno-Machado, L. (2002), 'Logistic regression and artificial neural network classification models: a methodology review', *Journal of biomedical informatics* **35**(5-6), 352–359.

ECB (2018), 'Fifth report on credit card fraud, september 2018'.

Erkmen, B. and Yildirim, T. (2008), 'Improving classification performance of sonar targets by applying general regression neural network with pca', *Expert Systems with Applications* **35**(1), 472 – 475.
URL: *http://www.sciencedirect.com/science/article/pii/S0957417407002667*

Ertekin, S., Huang, J., Bottou, L. and Giles, L. (2007), Learning on the border: active learning in imbalanced data classification, *in* 'Proceedings of the sixteenth ACM conference on Conference on information and knowledge management', pp. 127–136.

Estabrooks, A., Jo, T. and Japkowicz, N. (2004), 'A multiple resampling method for learning from imbalanced data sets', *Computational intelligence* **20**(1), 18–36.

Everett, C. (2009), 'Credit card fraud funds terrorism', *Computer Fraud and Security* .

Forman, G. H. (2006), 'Automated machine-learning classification using feature scaling'.

Garcia-Swartz, D. D., Hahn, R. W. and Layne-Farrar, A. (2006), 'The move toward a cashless society: a closer look at payment instrument economics', *Review of network economics* **5**(2).

García, V., Mollineda, R. A. and Sanchez, J. S. (2010), Theoretical analysis of a performance measure for imbalanced data, *in* '2010 20th International Conference on Pattern Recognition', IEEE, pp. 617–620.

García, V., Sánchez, J. and Mollineda, R. (2012), 'On the effectiveness of preprocessing methods when dealing with different levels of class imbalance', *Knowledge-Based Systems* **25**(1), 13 – 21. Special Issue on New Trends in Data Mining.
URL: *http://www.sciencedirect.com/science/article/pii/S0950705111001286*

Hastie, T., Tibshirani, R. and Friedman, J. (2009), *The elements of statistical learning: data mining, inference, and prediction*, Springer Science & Business Media.

He, H. (2013), *Imbalanced Learning: Foundation, Algorithms and Applications*, Wiley.

He, H. and Garcia, E. A. (2009), 'Learning from imbalanced data', *IEEE Transactions on knowledge and data engineering* **21**(9), 1263–1284.

Hoffmann, A. O. and Birnbrich, C. (2012), 'The impact of fraud prevention on bank-customer relationships: An empirical investigation in retail banking'.

Howley, T., Madden, M. G., O'Connell, M.-L. and Ryder, A. G. (2005), The effect of principal component analysis on machine learning accuracy with high dimensional spectral data, *in* 'International Conference on Innovative Techniques and Applications of Artificial Intelligence', Springer, pp. 209–222.

James, G., Witten, D., Hastie, T. and Tibshirani, R. (2017), *An Introduction to Statistical Learning with Applications in R*, Springer.

Japkowicz, N. et al. (2000), Learning from imbalanced data sets: a comparison of various strategies, *in* 'AAAI workshop on learning from imbalanced data sets', Vol. 68, Menlo Park, CA, pp. 10–15.

Jolliffe, I. T. and Cadima, J. (2016), 'Principal component analysis: a review and recent developments', *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **374**(2065), 20150202.

Kaggle (2017), 'Data set source'.
URL: *https://www.kaggle.com/mlg-ulb/creditcardfraud*

Kingma, D. P. and Welling, M. (2019), 'An introduction to variational autoencoders', *arXiv preprint arXiv:1906.02691* .

Krawczyk, B. (2016), 'Learning from imbalanced data: open challenges and future directions', *Progress in Artificial Intelligence* **5**(4), 221–232.

Krivko, M. (2010), 'A hybrid model for plastic card fraud detection systems', *Expert Systems with Applications* **37**(8), 6070–6076.

Lemaître, G., Nogueira, F. and Aridas, C. K. (2017), 'Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning', *Journal of Machine Learning Research* .
URL: *https://jmlr.org/papers/volume18/16-365/16-365.pdf*

Li, H. and Sun, J. (2012), 'Forecasting business failure: The use of nearest-neighbour support vectors and correcting imbalanced samples–evidence from the chinese hotel industry', *Tourism Management* **33**(3), 622–634.

López, V., Fernández, A., Moreno-Torres, J. G. and Herrera, F. (2012), 'Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. open problems on intrinsic data characteristics', *Expert Systems with Applications* **39**(7), 6585–6608.

Maaten, L. v. d. and Hinton, G. (2008), 'Visualizing data using t-sne', *Journal of machine learning research* **9**(Nov), 2579–2605.

Misra, S., Thakur, S., Ghosh, M. and Saha, S. K. (2020), 'An autoencoder based model for detecting fraudulent credit card transaction', *Procedia Computer Science* **167**, 254–262.

Mujalli, R. O., López, G. and Garach, L. (2016), 'Bayes classifiers for imbalanced traffic accidents datasets', *Accident Analysis & Prevention* **88**, 37–51.

Nguyen, G. H., Bouzerdoum, A. and Phung, S. L. (2009), 'Learning pattern classification tasks with imbalanced data sets', *Pattern recognition* pp. 193–208.

Schneider, F. (2013), 'The Shadow Economy in Europe'.

Siddhartha Bhattacharyya, Sanjeev Jha Kurian Tharakunnel, J. C. W. (2011), 'Data mining for credit card fraud: A comparative study', *Decision Support Systems* **50**, 602–613.

Snellman, J. S., Vesala, J. M. and Humphrey, D. B. (2001), 'Substitution of Noncash Payment Instruments for Cash in Europe', *Journal of Financial Services Research* pp. 131 – 145.

Sokolova, M. and Lapalme, G. (2009), 'A systematic analysis of performance measures for classification tasks', *Information Processing Management* **45**(4), 427 – 437.
URL: *http://www.sciencedirect.com/science/article/pii/S0306457309000259*

Sun, Y., Wong, A. K. and Kamel, M. S. (2009), 'Classification of imbalanced data: A review', *International journal of pattern recognition and artificial intelligence* **23**(04), 687–719.

Sweers, T., Heskes, T. and Krijthe, J. (2018), 'Autoencoding credit card fraud', *Bachelor Thesis* .

Zheng, A. and Casari, A. (2018), *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*, O'Reilly Media, Inc.