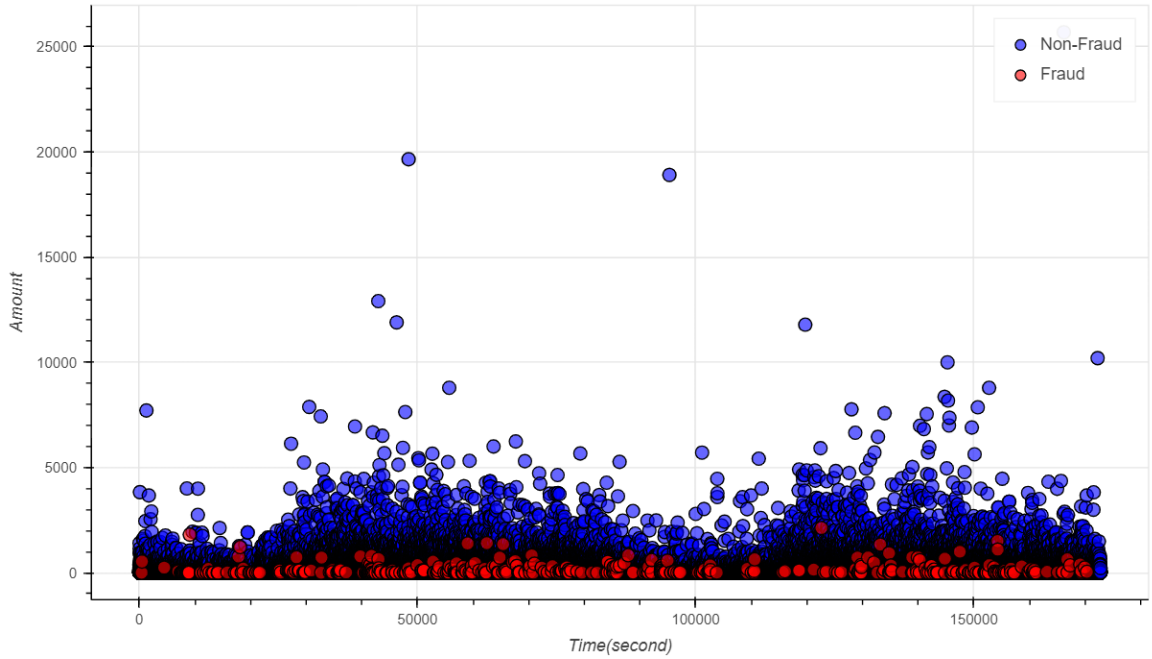
CHALMERS
UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG



Master's thesis

Credit Card Fraud Detection by Nearest Neighbor Algorithms

Master's thesis in Mathematical Sciences

Ramin Maghsood

Department of Mathematical Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2023

Master's thesis 2023

Master's thesis

Credit Card Fraud Detection by Nearest Neighbor Algorithms

Ramin Maghsood

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2023

Credit Card Fraud Detection by Nearest Neighbor Algorithms
Ramin Maghsood

Supervisor: Prof. Holger Rootzén, Department of Mathematical Sciences
Examiner: Prof. Serik Sagitov, Department of Mathematical Sciences

Master's Thesis 2023
Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Description of the picture on the cover page (if applicable)

Typeset in LaTeX
Gothenburg, Sweden 2023

Credit Card Fraud Detection by Nearest Neighbor Algorithms
Ramin Maghsood
Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg

Abstract

As the usage of internet banking and online purchases have increased dramatically in today's world, the risk of fraudulent activities and the number of fraud cases are increasing day by day. The most frequent type of bank fraud in recent years is credit card fraud which leads to huge financial losses on a global level. Credit card fraud happens when an unauthorized person uses another person's credit card information to make purchases. Credit card fraud is an important and increasing problem for banks and individuals, all around the world. This thesis applies supervised and unsupervised nearest neighbor algorithms for fraud detection on a Kaggle data set consisting of 284,807 credit card transactions out of which 492 are frauds, and which includes 30 covariates per transaction. The supervised methods are shown to be quite efficient, but require that the user has access to labelled training data where one knows which transactions are frauds. Unsupervised detection is harder and, e.g., for finding 80% of the frauds, the algorithm classifies more 50 times as many valid transactions as fraud cases. The unsupervised nearest neighbor distance method is compared to methods using the distance to the center of the data for fraud detection, and detection algorithms which combine the two methods. The L_2 distance and L_2 distance to zero and the combination of both distances are analyzed for unsupervised method. The performance of the methods is evaluated by the Precision-Recall (PR) curves. The results show that based on both area under curve and precision at 80% recall, L_2 distance to zero performs slightly better than L_2 distance.

Keywords: Fraud, Bank Fraud, Credit Card Fraud, Fraud Detection, Nearest Neighbor Algorithms

Acknowledgements

I would like to express my deep appreciation and sincere gratitude to my supervisor Prof. Holger Rootzén for his patience, encouragement and support. I would have never completed this thesis without his guidance and help. His great ideas and insightful comments were essential to me to finish my master's thesis.

I would like to thank my examiner Prof. Serik Sagitov for his support in this thesis. I would also like to thank Prof. Hjalmar Rosengren and Prof. Stefan Lemurell for their helpful support during my studies at the department of Mathematical Sciences.

Last but not least, I would like to thank my lovely parents and my great sister for their endless support and inspiration during my studies.

Ramin Maghsood, Gothenburg, March 2023

Contents

List of Figures

List of Tables

1
Introduction

In today's world, fraud is a global problem that leads to large financial losses all around the world. Developments in technology and digitization in recent years led to a massive increase in the number of fraudulent activities and in misusing the power of new technologies [7]. On the opposite side organizations and large companies are developing novel methods and effective solutions to detect fraud [4]. The recent developments in technology present new benefits and helpful methods to the banks, financial institutions and credit card issuers to decrease the risk of huge losses efficiently and to monitor and detect the fraud cases.

Millions of credit card transactions are processed every second and the humans are unable to analyse and process such amount of data to investigate the behavioral patterns of the fraudsters [13]. This is where the credit card fraud detection using machine learning algorithms play an essential role. Credit card fraud is divided into two types, online and offline fraud [18]. We provide more details about credit card fraud in section 2.3.

Credit card fraud detection is a challenging problem that banks and credit card issuers are struggling with and are making huge efforts in implementing fraud detection systems. In today's banking system, fraud detection is often done by using rule based methods. However, the progress and development of machine learning techniques gives banks and financial institutions the possibility to detect an unusual situation faster for big financial data sets.

The data set used in this thesis is the credit card transactions occurred in two days in Sep. 2013 and is downloaded from Kaggle [12]. Further information about the data set is provided in section 4.1. There is a number of authors that have done analysis on this data set and it has been widely studied and discussed further.

In [16] the author examined the three algorithms, Isolation Forest (IF), Local Outlier Factor (LOF) and One-Class SVM to identify the best performing unsupervised algorithms. The reason why the author used unsupervised approach for credit card fraud detection was the unavailability of the labelled data in the real world. In [17] the performance of supervised algorithms, K-Nearest Neighbor (KNN), Decision Tree (DT), Logistic Regression (LR) and Random forest (RF) were examined for credit card fraud detection and the author concluded that RF had the best performance for detection of fraud in credit card fraud transactions. In [18] the authors examined comparison of the performance of three classification methods Naive

Bayes, K-Nearest Neighbor and Logistic Regression which used for credit card fraud detection and their results showed that LR method performed better than the two others. More investigations of nearest neighbor algorithms for anomaly detection are [19],[20],[21].

In this thesis, we analyze the credit card transactions to do fraud detection. We investigate the function of nearest neighbor distance methods in credit card fraud detection and evaluate the performance of supervised and unsupervised methods in finding and detecting fraud cases in credit card transactions. The distance metrics used in the nearest neighbor methods are the L_2 distance, L_{30} distance, L_2 distance to zero, and inverse L_2 distance. We conclude the thesis with the pros and cons of supervised and unsupervised nearest neighbor distance methods.

2
Background

In this chapter, we give a general overview of anomalies and anomaly detection and of the different types of financial fraud. We focus on credit card fraud and investigate the different approaches used for credit card fraud detection.

2.1 Anomalies and Anomaly Detection

An anomaly is a deviation from normal pattern and recognizing an anomaly hence depends on how one defines a normal behavior [9]. Once normality is defined, it is much more easier to define what is an anomaly. Anomalies are commonly divided in the three types point anomaly, contextual anomaly and collective anomaly [10]. A point anomaly (the simplest type of anomaly) is a point which deviates from the majority of data points, a contextual anomaly is a point that is anomalous in regard to its neighbors, and a collective anomaly is a group of data points which behaves differently from other groups [3].

Anomaly detection is the process of identifying unusual data in a data set. It is an important technique that is applied in various domains [5]. In the banking and finance area, any set of unusual activities such as extraordinary transactions are referring to anomalies. Anomaly detection is highly applicable and useful in banks and financial services companies and is considered as a data analysis task in financial domains [6]. The anomaly detection in data sets with high dimensions is becoming one of the main financial problem. The issue of high dimensional big data affects the performance and the accuracy of the existing anomaly detection techniques [14].

2.2 Financial Fraud

Financial fraud is happening when another person takes your money or any other assets by means of a fraudulent activity [1]. In recent years, advancements in technology and digitization of the real world have led to an increase in the number financial fraud cases, and fraudsters misuse advances in technology to take advantage from fraudulent activities. The common categories of financial fraud are *bank* fraud, *corporate* fraud and *insurance* fraud [2]. Bank fraud occurs when any unauthorized transactions are made in your bank account. Corporate fraud mostly happens in forms of a financial statement fraud and securities and commodities fraud and insurance fraud can occur for any kind of insurance such as car and health care insurance [2]. The most common types of bank fraud are credit card fraud, money laundering,

and mortgage fraud [10]. Many of the investigations on financial fraud detection in recent years were conducted in the area of credit card fraud. More details about credit card fraud are given in the next section.

2.3 Credit Card Fraud

A credit card fraud happens when an unauthorized person uses another person's credit card information to make purchases. Credit card fraud is a big problem that affects many people all around the world [18]. Credit card fraud is mainly committed by a fraudster by means of obtaining the access to the cardholders physical credit card information [17]. Nowadays the payment card and credit card issuers are doing everything they can to protect their customers from the payment and credit card fraud. Credit card fraud can happen in the form of online purchases or in the shape of misusing someone else card for shopping in the stores [16]. As a credit card holder, it is normal to be concerned about the fraudulent activities that may happen. On the other side, credit card issuers are also concerned about this matter because in many cases they have to pay back the loss to their customers.

Credit card issuers use rule based methods and different ways to do fraud detection. One way is to employ the use of highly advanced fraud detection software. The software check the transactions and based on prior information determine whether the transaction is fraudulent or is a reliable transaction. The other way used by credit card issuer is to look for patterns used by the credit card holders which means that if the card holder always uses the card in a similar manner but suddenly a transaction falls out of the card holder's normal pattern, triggers the credit card company try to investigate whether that transaction is valid or not [11].

Many people use their credit card to pay for lunch at a restaurant or to make a purchase at a store while they are out of the city and these payments are out of their normal pattern of using the credit card for purchases. In some cases, they might be asked by their bank to validate credit card transactions and to verify whether they have made those transactions at these particular locations or not. It is normal that the banks are obliged to follow their rule-based regulations in particular situations. In many cases, these transactions are false positives which means that they are exactly the person used the credit card at the corresponding restaurant they have never gone there before and made the purchase at the store they have never gone there before. And in these situations, the customers are asked by the credit card issuer to verify this matter and to confirm that the transactions is accurate or not. If some of those transactions are invalid or be fraudulent then obviously the card holder must have to let the credit card issuer know about and ask them to shut the account down as early as possible and issue a new credit card [11].

3

Theory

In this chapter, we describe machine learning approaches used in credit card fraud detection and in particular those used in this thesis. As we use a financial data set, it is highly important to find out which method can be more effective and appropriate for financial fraud detection and whether the used method is working properly with the chosen data set or not. We concentrate on supervised and unsupervised nearest neighbor distance algorithms and on the theory behind them. In supervised anomaly detection, on a training set one knows what is an anomaly and what is not an anomaly whereas in unsupervised anomaly detection, one doesn't know it. To evaluate unsupervised anomaly detection methods, one needs to have labeled data. Labeled data is data that one knows whether it is anomaly or not anomaly.

3.1 Supervised Machine Learning

Supervised learning is the process of classifying a new data point where the labelled data is available. In other words, it uses labeled data to be able to train the models for classification of a new data set. For example in areas that classification approaches used for anomaly detection, labeled data means that we know which instances are anomalies.

Some of the examples of classification algorithms used in fraud detection are K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree (DT) and Random Forest (RF).

The KNN algorithm is one of the simplest machine learning classifiers. In this algorithm, a data point is classified by its nearest neighbors. The parameter k is the number of nearest neighbors and a new point is classified based on the label of the majority of its neighbors [29]. In paper [20] the KNN was presented as a precise method in decreasing the number of false alert and detecting fraudulent transactions and was introduced as an efficient algorithm for credit card fraud detection.

The Support Vector Machine algorithm was first presented by Vapnik [23]. The method is a discriminative algorithm to find an optimal hyperplane (a decision boundary in binary case) for separating the data space for a given labeled data set [31]. The use of SVM as a credit card fraud detection technique in high dimensional data sets is analysed in paper [39] and the authors concluded that this algorithm gives better results while using small data sets.

The Decision Tree algorithm is another supervised learning method which is popular and highly used for prediction and classification problems. A decision tree is a classification tree with main components of *root* node, *decision* node and *leaf* node. The root node in the algorithm is represented as the starting point, the decision node in the algorithm is a point where deciding to split the tree, and the leaf node in the algorithm is denoted as the final decision [24]. An investigation on credit card fraud detection by decision tree methods and a comparison of its performance with SVM is presented in [25]. The author concluded that Decision Tree algorithms perform better than SVM as the number of frauds detected by this algorithm are far more than the other.

The Random Forest algorithm was introduced by L. Breiman [26]. It is a tree-based algorithm used for classification and regression problems. The algorithm is constructed from multiple decision trees and reached to a decision by maximum number of votes [17]. A well-written investigation of random forest is presented in [27]. The author investigated the weak performance of this algorithm in detecting anomalies.

3.2 Unsupervised Machine Learning

The unsupervised learning method is similar to classification but does not use labelled data. The unsupervised based anomaly detection technique is one of the methods used to investigate the concept of clustering for financial fraud detection and different models of clustering techniques examined in [6]. The unsupervised methods do not need the prior knowledge of fraudulent and non fraudulent transactions in historical database, but instead detect changes in behavior and patterns caused by unusual transactions. Some examples of unsupervised learning algorithms are K-means, One-Class SVM, Isolation Forest (IF), and DBSCAN which are used for anomaly detection.

The k-means algorithm is one of the most commonly used clustering algorithms which is classifying the data points into the pre-specified number of classes, say k. The algorithm starts by choosing k points in random as the centroids of classes, and assigning the points to the closest cluster, then for each cluster the mean of all points assigned to that cluster will be calculated and considered as the new centroid and the previous steps will be repeated until no points change cluster membership [29]. In paper [8] the clustering techniques used for financial fraud detection were reviewed.

One-Class SVM is an unsupervised machine learning algorithm used for anomaly detection. The main idea behind this algorithm is to classify the data set into binary classes [16]. The one-class SVM is highly applicable and useful for imbalanced data set where there are a large number of normal data and a few number of anomaly data [32]. In paper [38] the authors analysed the one-class SVM as one of the best methods used for binary class data sets.

Isolation Forest is an unsupervised algorithm which was invented by Fei Tony Lui in 2007 [13]. The methodology in IF algorithm is the same as Random Forest and the construction of algorithm is on the concept of Decision Trees [16]. A deep analysis of IF and the comparison of its algorithm with other unsupervised learning methods is presented in [33].

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density based clustering algorithm. The algorithm categorizes all data points to core points, border points and noise points where the core points represent the minimum number of points that should be considered in each cluster, the border points represent the points that are reachable from or via other core points, and the noise points represent those points which are not core or border points [28].

In the next section, we study the Nearest Neighbor Distance Methods and discuss more that why it would be a good approach for credit card fraud detection.

3.3 Nearest Neighbor Anomaly Detection

The nearest neighbor methods are the models that find a number of training sample points that are closest to a test point and forecast the label of the test point. The number of training points in the model can be a constant k or can be varied based on the local density of points and the most common distance metric used is the standard Euclidean distance [36]. The nearest neighbor distance method can be divided into the supervised and unsupervised versions.

3.4 The Supervised Nearest Neighbor Method

In the supervised method we looked at the label of closest neighbor of each point from test data set and used it as the predicted label for the test point. According to the algorithm, a data point is classified based on how its neighbors are classified. In addition to the closest neighbor, we also consider the first two, the first three and the first four closest neighbors to predict the labels by assigning a fraud label to a test data point if at least one of its neighbors has fraud label.

3.5 The Unsupervised Nearest Neighbor Method

In the unsupervised setting, we pretend that the true labels are not available and look for ways to classify the test data points as fraud and non-fraud cases. For this, we set a threshold as a cut-off point and classified all test points above that threshold as fraud cases. The assumption is that the fraud cases should get the highest average distances.

Assume that d_i is the distance of the test point i to its closest neighbor in training set. Then for a given cut-off value c, we predicted the class label as follows:

$$\hat{Y}_i = \begin{cases} 1 & \text{if } d_i \geq c \\ 0 & \text{if } d_i < c \end{cases} \tag{3.1}$$

The distance d_i can be any $L2$, $L30$ norms or $L2$ distance to zero. It can be also the average distance of the k closest neighbors of point i.

3.6 Evaluation Metrics

There exists different methods to evaluate the performance of an algorithm used for fraud and anomaly detection. In this thesis, we mainly focus on the standard evaluation metrics Precision, Recall and Accuracy. These metrics are defined based on true positive, true negative, false positive and false negative rates as

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where

- True Positive (TP): The predicted label is fraud while the true label is fraud.
- True Negative (TN): The predicted label is non-fraud while the true label is non-fraud.
- False Positive (FP): The predicted label is fraud while the true label is non-fraud.
- False Negative (FN): The predicted label is non-fraud while the true label is fraud.

High precision means we got low false positive rate and high recall means we got low false negative from the algorithm. The precision measures the ratio between the true positives and all the positives, and the recall measures the ratio of correctly true positives identified by the algorithm. On the other hand, accuracy is the ratio between the total number of correct predictions and all predictions. The precision and recall values can be presented by precision-recall curve. The precision-recall curve is a visualization of the precision values against the recall values computed for different thresholds. It is a useful measure for a data set with imbalanced classes [37].

Another way of showing the performance of the algorithm is given by the confusion matrix, which is a cross-table for comparing the predicted labels with the true ones. Table 3.1 shows the confusion matrix for the Non-fraud which is considered as negative class and the Fraud which is considered as positive class [13],[27].

Table 3.1: Confusion matrix for non-fraud and fraud

		Predicted	
		Non-Fraud	Fraud
Actual	Non-Fraud	True Negative (TN)	False Positive (FP)
	Fraud	False Negative (FN)	True Positive (TP)

4

Methods

In this chapter, we present an analysis on the Kaggle data set. In our analysis, we look at different portions of training and test sets. First we consider the first 20% of the data set as training and the next 10% as test set. Then we look at the first 80% of the data as training and the rest 20% as test set. Finally the first 20% of the data as training and the rest 80% as test data sets are chosen.

4.1 Data set

The data set used in this thesis consists of 284,807 credit card transactions, where 492 are fraudulent. It includes twenty eight features V1,..., V28 obtained from the original features using principal components analyses (PCA). The non-transformed features in this data set are "Time", "Amount", and "Class". The transactions are labeled to fraud (1) and non-fraud (0) cases and presented in the feature Class as the target variables. There are no missing values in the data set [12].

4.2 Exploratory data analysis

As V1, ..., V28 are anonymous, we focus on two main features Time and Amount and have a deeper look at them in this section. We represent the time histogram of fraud and non-fraud transactions and visualize the scatter plot of amount over time.

Time feature

The Time feature is the number of seconds between the first transaction and the current transaction in the data set. Figure 4.1 shows the distribution of time feature over two days. It can be seen most of the transactions are made during the day.

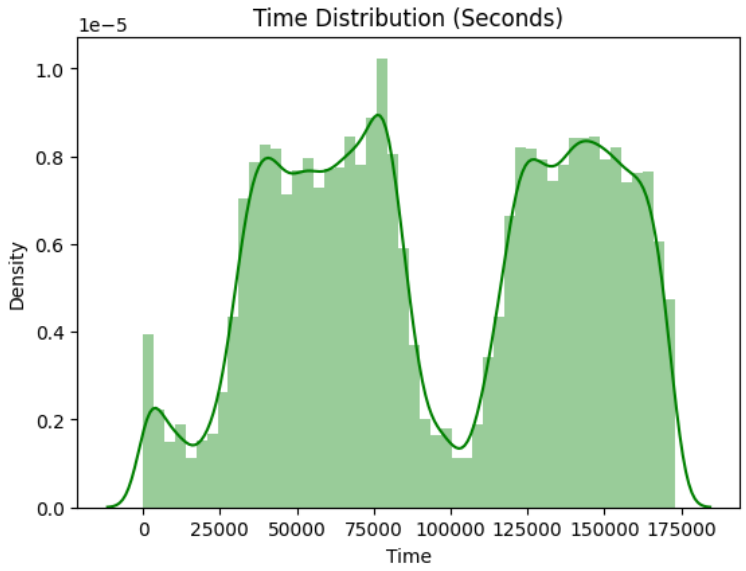


Figure 4.1: Distribution of time feature

To better understand if the transactions occurred at unusual times, the time histogram of fraud and non-fraud transactions is displayed separately in figure 4.2.

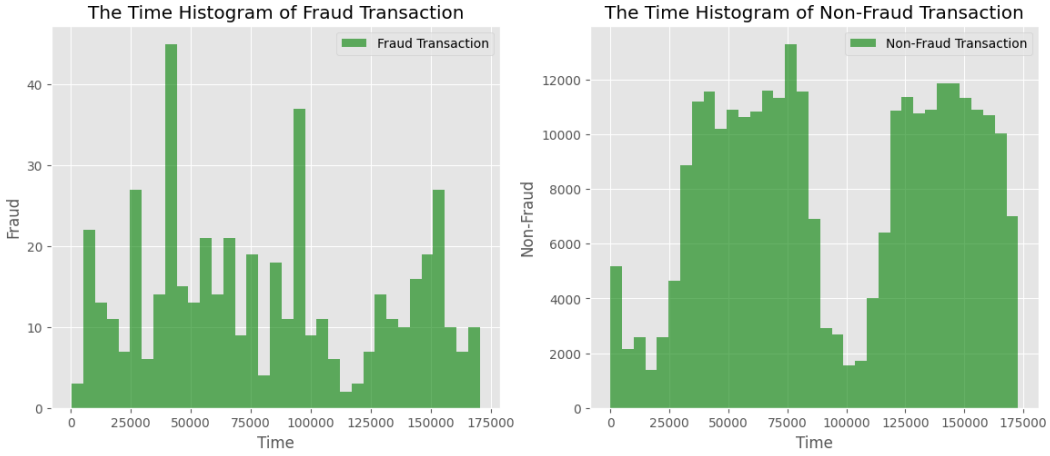


Figure 4.2: Time histogram of fraud and non-fraud transactions

We can see that the time feature is not distinguishing the fraud and non-fraud cases very well. However, one can say that the number of fraud transactions are higher during night hours.

Amount feature

The Amount feature is denoted as the amount of transaction in the data set. The scatter plot of amount over time is shown in figure 4.3. It can be seen that there are frauds only on the transactions with the amount less than 2500.

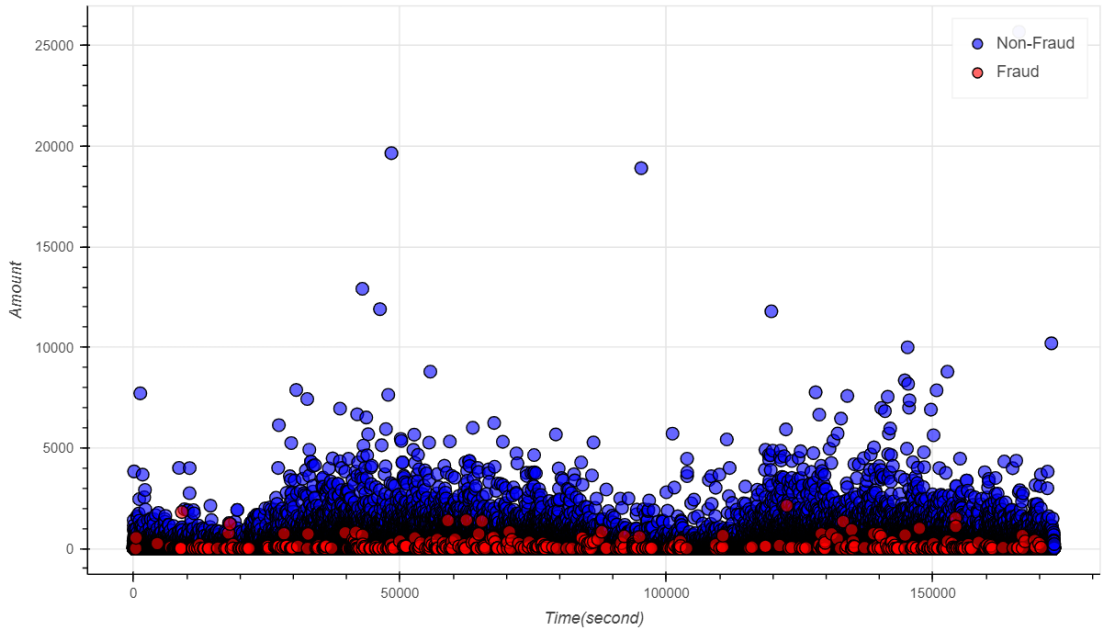


Figure 4.3: Scatter plot of amount over time

The distribution of amount feature and its logarithm are shown in figure 4.4 shows for non-fraud transactions and in figure 4.5 for fraud transactions.

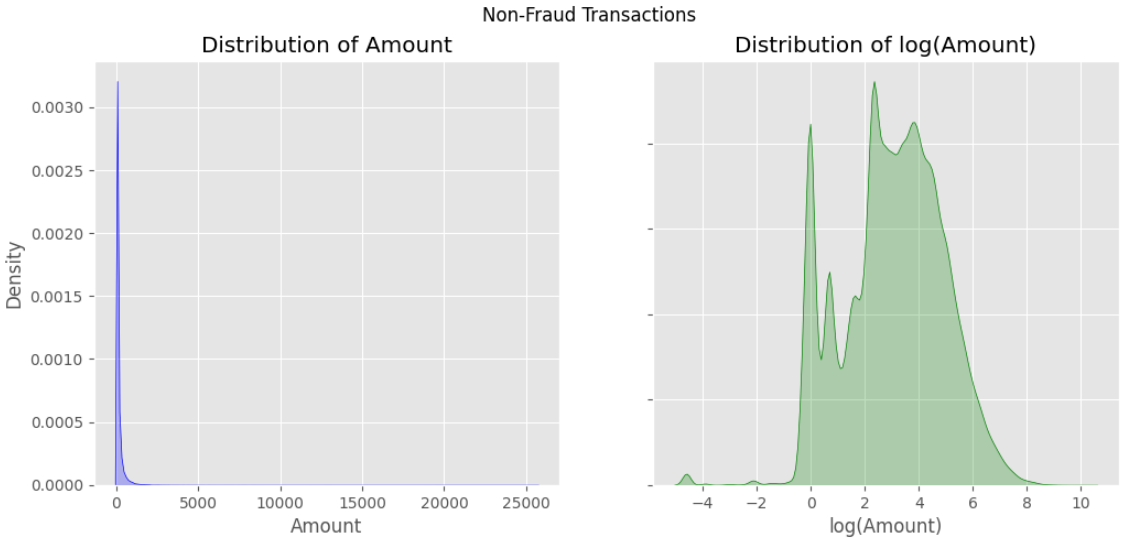


Figure 4.4: Distribution of amount feature (left) and its logarithm (right) for non-fraud transactions

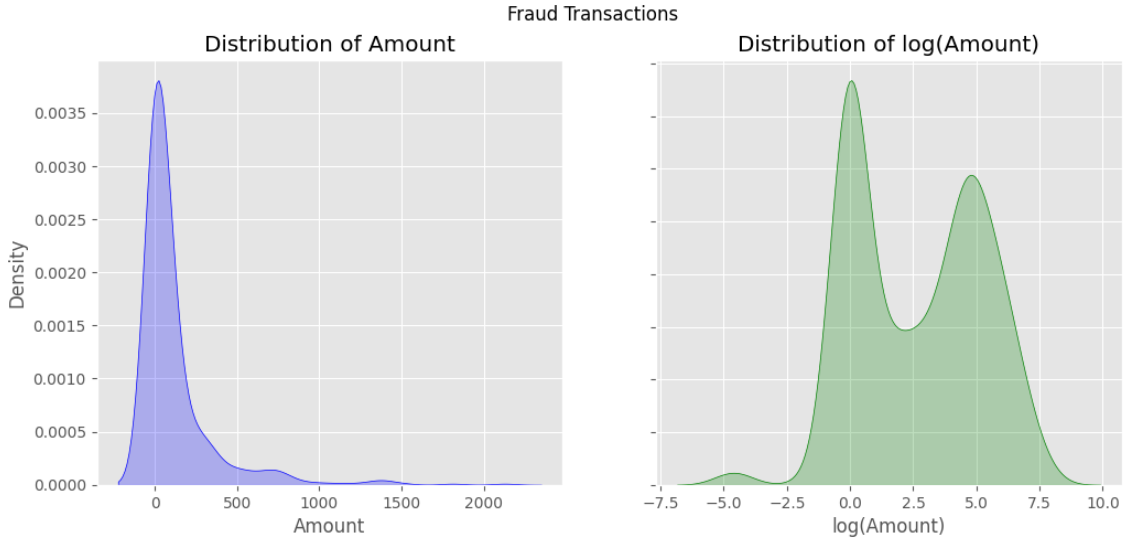


Figure 4.5: Distribution of amount feature (left) and its logarithm (right) for fraud transactions

Table 4.1 shows the difference in the amount of money used in different transaction classes. By looking at the table, one can see that 75% of fraud cases have the amount around 105. This indicates that the fraudsters make lots of low amount transactions frequently, which make it hard to find them for credit card issuer.

Table 4.1: Amount details of fraud and non-fraud transactions

Amount	Fraud	Non-Fraud
Count	492	284315
Mean	122.21	88.29
std	256.68	250.11
min	0.00	0.00
25%	1.00	5.65
50%	9.25	22.00
75%	105.89	77.05
max	2125.87	25691.16

4.3 Training and Test Set

For further analysis, we first consider a small portion of the data set where we have 20% and 10% for training and test respectively, then analyse the results for 80% as training set and 20% as test set and the case where the training set is 20% and the test set is 80%. The main reason for dividing the data set into the different portion of train and test sets corresponds to different practical situations. It also depends on how quickly one can have access to the data. For some investigators the time of detecting fraud situations is matter and it might be better for them to take the already existing 20% of labeled data as training and do the prediction for the next small available 10% of the data. On the other hand, in some cases it might be better

to wait until getting the 80% of labeled data be ready and then do the investigation for the rest of the data.

4.4 Standardizing Variables

In order to do the analysis on the data set, we need to define appropriate distance metrics. To compute the distances correctly we need to make the range of all variables are similar. Hence, all the variables need to be standardized. Here a variable that is standardized has a mean of 0 and a standard deviation of 1. To do the standardizing, we subtract the mean from the variable and then divide it by the standard deviation. By considering the number of credit card transactions as the number of observations x_{ij}, for $i = 1, 2, ..., N$ where $N = 284806$ and the number of features, $j = 1, 2, ..., n$ where $n = 30$ is the number of columns. For each column, the mean is calculated by

$$\overline{x}_{.j} = \frac{\sum_{i=1}^{N} x_{ij}}{N},$$

and the standard deviation is calculated as the following,

$$\sigma_{.j} = \sqrt{\frac{\sum_{i=1}^{N}(x_{ij} - \overline{x}_{.j})^2}{N-1}}.$$

Then, the standardized column variables are obtained as [13]

$$x^*_{ij} = \frac{x_{ij} - \overline{x}_{.j}}{\sigma_{.j}}.$$

Let x be an n-dimensional vector, x=(x_1,x_2,...,x_n). The Lp-Norm, written as $||x||_p$ for p>0, is defined as

$$||x||_p = (\sum_{i=1}^{n} |x_i|^p)^{1/p}.$$

The exponent p is commonly chosen to be 2 or infinity. In our analysis, we measured the distances based on $p = 2$ as the standard metric and used $p = 30$ as an approximation to $p = \infty$ as it is easier to handle the computations.

4.5 Computation of Nearest Neighbor Distances

In this section, we analyse three algorithms used in the nearest neighbor search to find the closest data point to a test data point. The three algorithms are *Brute Force*, *KD Tree* and *Ball Tree*.

Brute Force algorithm is the most accurate nearest neighbor method which considers all data points and the efficiency of algorithm in small data sets is feasible

however it has computational inefficiency in data sets with high dimensions [35]. In this method, one calculates the distance between a test data point and each point in the training set and the point that has the lowest distance with the test point is considered as the nearest neighbor [34]. The brute force could be an ideal method but due to the computational complexity and large memory requirements, it would be better to compare the results with other nearest neighbor methods.

KD Tree is a commonly used nearest neighbor algorithm where the data points are divided into two sets at each node and it is a binary tree algorithm which finally ends in two nodes [35]. The first step in this method is to pick randomly a feature (depends on the number of features one has in the data set) then try to find the median in the features, and split the data set into the same halves, then after picking the next feature, repeat the previous steps, and at last continue the procedure until partitioning all the data points [34].

Ball Tree algorithm is in a shape of a metric tree and in this algorithm the data points are divided into two clusters where each cluster is surrounded by a circle [35]. The first step in this method is to take a random point, then by finding the farthest data point from that random point, try to find the data point which is farthest to this farthest point, then considering a line between the two farthest points and put all the data points on this line, and trying to find the median in order to split the space into two halves, then find the centroid in each half and look for the farthest data point to the centroids and then drawing a circle around the centroid with the farthest distance as a radius and continue repeating the process within the remained circles [34].

We tried three methods Ball Tree, KD Tree and Brute force with two metrics L_2 and L_{30} for distance calculations.

Table 4.2: Running time for computing L_2 nearest neighbor distances for 3 different algorithms

Time (seconds)	10%	20%	50%
Ball Tree	64.68	307.87	2014.46
KD Tree	37.55	125.33	477.23
Brute Force	0.87	2.78	21.56

Table 4.2 shows the interesting fast that the Brute Force was substantially faster than the other methods.

Table 4.3: Running time for computing L_{30} nearest neighbor distances for 3 different algorithms

Time (seconds)	10%	20%	50%
Ball Tree	706.14	2803.28	16958.16
KD Tree	175.12	444.85	1038.55
Brute Force	103.16	395.07	2582.08

By Table 4.3 the Brute force algorithm was fastest for the 10% and 20% splits but KD Tree was faster for the 50% data.

However, since the Brute force algorithm doesn't use any approximations, and since it was possible to use also in the case where KD Tree was faster, we hereby used the Brute force algorithm to find the nearest neighbor distances.

We have throughout used Python programming language for the data analysis. The main used packages are "numpy", "pandas", "scipy" and "sklearn". The main class that we used for nearest neighbor search was "NearestNeighbors" [36]. Table 4.4 shows the computer specifications and the programming language used for data analysis.

Table 4.4: Hardware and software specifications for the PC

Operating System	RAM	CPU	Program
Windows 10	16GB	2.3GHz Intel Core i7	Python 3.10.7

4.6 Supervised and Unsupervised Anomaly Detection using Nearest Neighbor Distance

The nearest neighbor distance algorithm is one of the anomaly detection approaches. In this thesis, an anomaly is referring to a fraudulent transaction. We used both the supervised and unsupervised nearest neighbor distance methods for fraud detection.

In the supervised nearest neighbor distance algorithm, one finds the closest point in the training data set to each test data point, and assign the label of this point to the test data point. We then compared the assigned labels with their true labels to see how well the method worked. In addition to the first closest neighbor, we also considered using the first two, the first three and the first four closest neighbors to predict the test labels. For this, we assigned label 1 (fraud) to a test data point, if at least one of its neighbors had label 1. Then we evaluated the false positive and false negative errors for each of those methods.

In unsupervised learning nearest neighbor algorithm, one can pretend that the class labels are not available and try to look at the distances to the nearest neighbors for clustering the data. In the unsupervised method, we take a point from the test data set and find its nearest neighbor in the training set. One compute the L_2 distance between the test point and its nearest in the training set to predict whether the point in the test set is anomalous or not. If the distance is very large or very small then this could be an indication that the point is anomaly. As mentioned in Section 3.5 we set a threshold for distances and predict the label as fraud if the distance of a test data point is larger than the threshold. The results from supervised and unsupervised methods are presented in sections 5.1 and 5.2.

4.7 Combining different anomaly detection methods

Another possibility we have considered in unsupervised analysis is to combine L_2 distance and L_2 distance to zero to predict fraud and non-fraud. In order to do this, for each point i from test set we first computed the L_2 distance d_{1i} to the closest neighbor from training set. Then, the L_2 distance to zero d_{2i} is computed for the same point i from test set. For each distance, we set a cut-off point range between 10% and 99.9% quantiles, defined as $c_1 = (a_1 - b_1)$ for L_2 distance and $c_2 = (a_2 - b_2)$ for L_2 distance to zero. Finally, we combined the cut-off points and defined a common range for cut-off points between $c = (mean(a_1, a_2) - mean(b_1, b_2))$, and we predicted the class label as follows:

- If at least one of the distances is greater than the cut-off point

$$\hat{Y}_i = \begin{cases} 1 & \text{if } d_{1i} \geq c \text{ or } d_{2i} \geq c \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

- If both distances are greater than the cut-off point

$$\hat{Y}_i = \begin{cases} 1 & \text{if } d_{1i} \geq c \text{ and } d_{2i} \geq c \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

The two distances d_{1i} and d_{2i} are not comparable due to different distributions they have. If one has 'OR' condition, then it is a big distance which dominates. If one has 'AND' condition, then it is a small distance which dominates. Therefore, it is good to make transformation of the distances to have the same scale for the combination. In our analysis, we used standardization and uniform transformation of the distances. We replace d_{1i} and d_{2i} in Equations 4.1 and 4.2 once with their standardized values and once with their uniform transformed values. Then the fraud cases are predicted by combining the two distances.

The standardization is subtraction of mean and division by standard deviation. The transformation to uniform distribution can be defined as follows. If x is a continuous random variable with distribution function $F(x)$, then $F(x)$ can be estimated from a sample x_1,x_2,...,x_n as

$$\hat{F}(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\{x_i \leq x\}}$$

where

$$\mathbf{1}_{\{x_i \leq x\}} = \begin{cases} 1 & \text{if } x_i \leq x \\ 0 & \text{if } x_i > x \end{cases}$$

Thus

$$Y_1 = \hat{F}(x_1), ..., Y_n = \hat{F}(x_n)$$

has approximately a uniform distribution on [0,1].

5

Results

In this chapter, we describe the results obtained based on the nearest neighbor distance algorithms. This includes both supervised and unsupervised learning approaches. In the supervised section, the analyse of L_2 norm and L_{30} norm is described while in the unsupervised section, the L_2 distance and L_2 distance to zero are analysed. In this thesis, the L_2 distance means the distance to the nearest neighbor and the L_2 norm is referring to the norm of this distance. The analysis is done on the three data splits of 20/10, 80/20 and 20/80 for the training and test sets.

5.1 Supervised Nearest Neighbor Distance

In this section, the results of the supervised nearest neighbor distances based on L_2 norm and L_{30} norm are presented. This includes the confusion matrix of the nearest neighbor algorithm, and the comparison table of true positive, false positive, and false negative errors.

5.1.1 L_2 norm

The nearest neighbor distance algorithm is used based on L_2 norm for the three data splits of training and test sets. For each point from the test set, we found the closest point from the training set based on the minimum distance. The histogram of the distances for three portions of test sets are shown in Figure 5.1

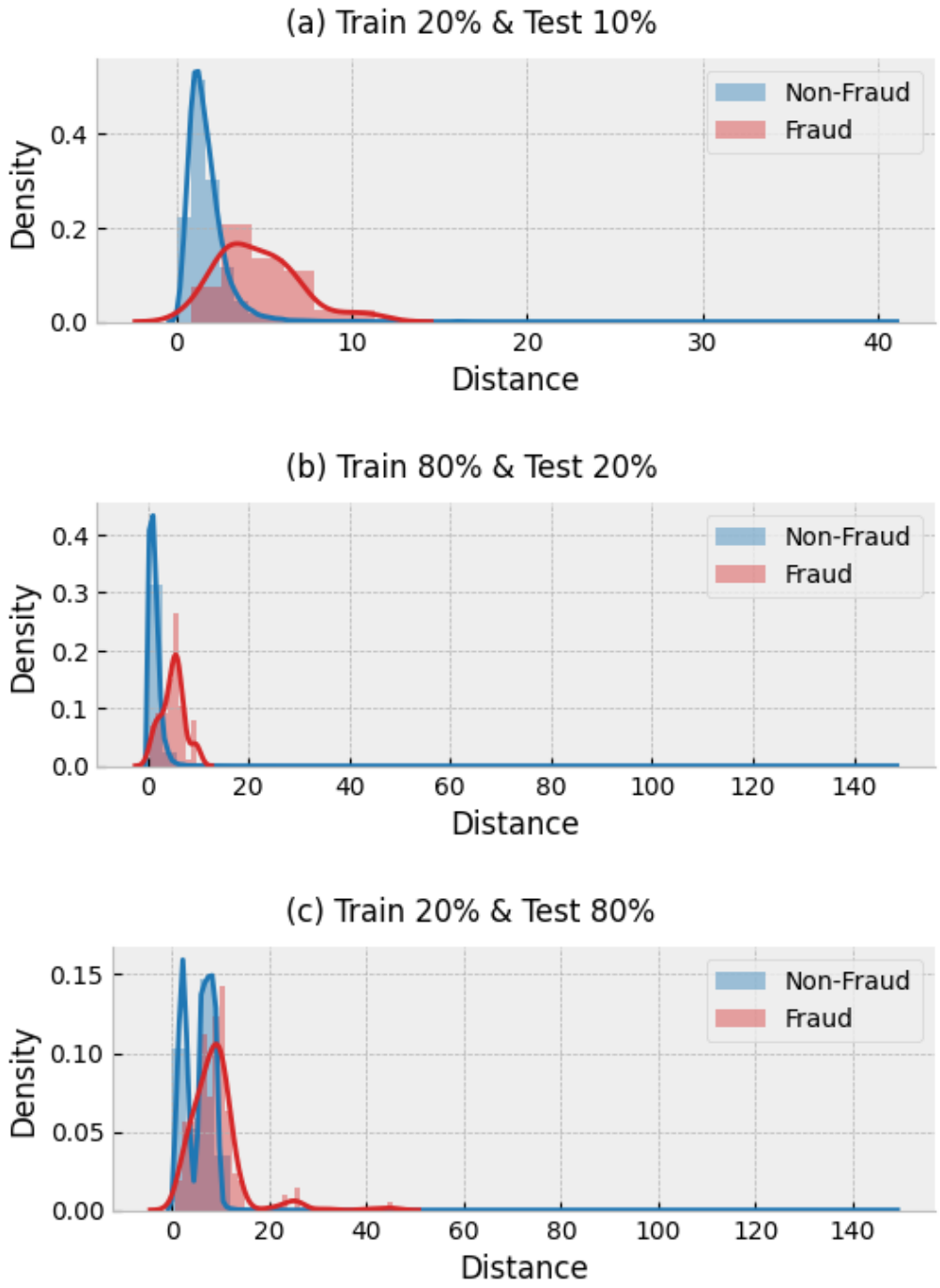


Figure 5.1: Histogram of distances based on L_2 norm. (a) For 20% train and 10% test sets, (b) For 80% train and 20% test data sets, (c) For 20% train and 80% test data sets.

To know how many of fraud cases got a non-fraud nearest neighbor, we compared the label of closest neighbor of each point with their true labels. The confusion matrix for three different splits of data set for the first closest neighbor based on L_2 norm is shown in Table 5.1 .

(a)

		Neighbor Class	
		Non-Fraud	Fraud
Class	Non-Fraud	28422	12
	Fraud	13	34

(b)

		Neighbor Class	
		Non-Fraud	Fraud
Class	Non-Fraud	56880	7
	Fraud	21	54

(c)

		Neighbor Class	
		Non-Fraud	Fraud
Class	Non-Fraud	227230	281
	Fraud	101	234

Table 5.1: Confusion matrix of the nearest neighbor algorithm with L_2 norm. (a) For 20% train and 10% test sets, (b) For 80% train and 20% test data sets, (c) For 20% train and 80% test data sets.

In addition to use the first closest neighbor for predicting the labels of test data points, we also looked at the first two, the first three and the first four closest neighbors to predict the test labels. As we mentioned before, we assigned label 1 (fraud) to a test data point, if at least one of its neighbors has label 1.

The comparison of true positive, false positive and false negative errors for three data splits is shown in Table 5.2. As can be seen in this table, the false positive error, which is the number of incorrectly assigned non-fraud labels to the fraud cases, is increasing while we are looking at the label of far neighbors.

(a)

	First Neighbor	Second Neighbor	Third Neighbor	Forth Neighbor
True Positive	34	36	36	36
False Positive	12	27	33	41
False Negative	13	11	11	11

(b)

	First Neighbor	Second Neighbor	Third Neighbor	Forth Neighbor
True Positive	54	55	57	57
False Positive	7	20	28	41
False Negative	21	20	18	18

(c)

	First Neighbor	Second Neighbor	Third Neighbor	Forth Neighbor
True Positive	234	241	243	247
False Positive	281	501	730	956
False Negative	101	94	92	88

Table 5.2: Comparison of true positive, false positive and false negative between the first four neighbors based on L_2 norm. (a) For 20% train and 10% test sets, (b) For 80% train and 20% test data sets, (c) For 20% train and 80% test data sets.

5.1.2 L_{30} norm

Since there are 30 features in our data set, we measured the distances based on L_{30} norm as an approximation of L_∞ to handle the computations easier. The results of L_{30} are pretty similar to L_2 norm and even a little bit worse. The running time for L_{30} was too much, so we just presented the outputs for the small portion of the data set where we have 20% for training and 10% for test.

The same as before, for each point from test set, we found the closest point from the training set based on the minimum distance. The histogram of the distances based on L_{30} norm for 20% train and 10% test data sets is shown in Figure 5.2.

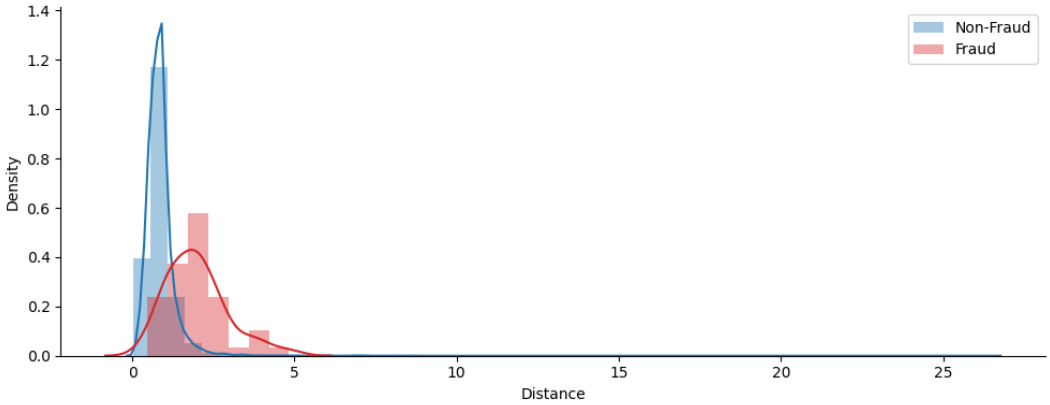


Figure 5.2: Histogram of distances based on L_{30} norm for 20% train and 10% test data sets

To know how many of fraud cases got a non-fraud closed neighbor, we compared the label of closest neighbor of each point with their true labels. The Table 5.3 indicates the confusion matrix for the first closest neighbor based on L_2 and L_{30}.

		Neighbor Class	
		Non-Fraud	Fraud
Class	Non-Fraud	28417	17
	Fraud	12	35

Table 5.3: Confusion matrix of the nearest neighbor algorithm with L_{30} norm for 20% train and 10% test data sets

It can be seen that the performance of L_{30} is not as good as L_2 and we have more false positive value.

	First Neighbor	Second Neighbor	Third Neighbor	Forth Neighbor
True Positive	35	36	36	36
False Positive	17	36	46	59
False Negative	12	11	11	11

Table 5.4: Comparison of true positive, false positive and false negative between the first four neighbors based on L_{30} norm for 20% train and 10% test data

5.1.3 Summary

By looking at the tables 5.2 (a-c), we can see that if k is getting larger, the false negative errors go down and the false positive errors go up. As the recall value is presenting the true positive rate, we computed the recall values for the three data splits (20/10, 80/20 and 20/80) to be 72.3%, 72% and 69.8% respectively and the precision values to be 73.91%, 88.52% and 45.43% respectively.

It is more important to have high recall than having high precision for fraud detection as one doesn't want to miss fraud cases. By considering the recall values of 20/10 and 80/20, we can conclude that the small portion of 20/10 was performing better than the other splits.

We also computed the accuracy measurements for the three splits and achieved the following percentages 99.91%, 99.95% and 99.83% respectively however the accuracy here doesn't tell anything about the performance, since we have very many non-fraud cases and very few fraud cases, therefore, in compare with TN, the values of TP, FP and FN are so small so that the Accuracy will be nearly equal to one as TN/TN=1.

In general, We can conclude that the supervised methods are quite efficient but require that the user has access to labelled training data where one knows which transactions are frauds.

5.2 Unsupervised Nearest Neighbor Distance

We describe here the unsupervised nearest neighbor L_2 distance algorithm and the L_2 distance to zero algorithm applied to the 20/10, 80/20 and 20/80 subsets from training and test sets splits. We also present the combination of L_2 distance and L_2 distance to zero in this section. In addition, the inverse L_2 distance is used as another metric for fraud detection.

5.2.1 L_2 distance

The nearest neighbor distance algorithm is used based on L_2 norm for different portions of training and test sets. For each point from test set, we found the closest point from the training set based on the minimum distance. The scatter plot of distances based on $L2$ norm is shown in Figure 5.3.

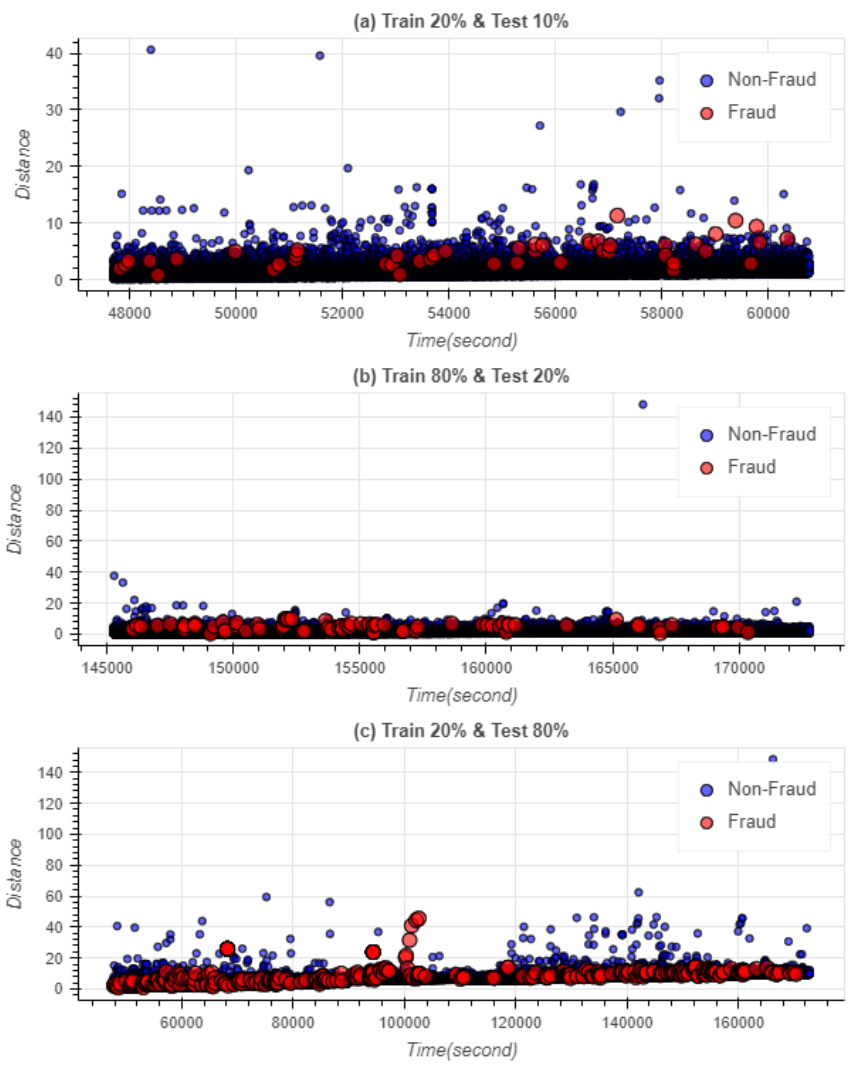


Figure 5.3: Scatter plot of distances based on L_2 norm. (a) For 20% train and 10% test sets, (b) For 80% train and 20% test data sets, (c) For 20% train and 80% test data sets.

As can be seen in the plots, the fraud and non-fraud cases are not separating very well and we have some non-fraud cases with a very large distance.

To classify and predict the test data labels based on unsupervised learning algorithm, we used Equation 3.1. We set different values for the cut-off point which varied between 10% and 99.9% quantiles of L_2 distance, and predicted the label of test data points. To be able to see the performance of the algorithm in a better way, we computed the precision and recall values. The Precision-Recall curve based on L_2 norm is shown in Figure 5.4 for three different portions of training and test sets.

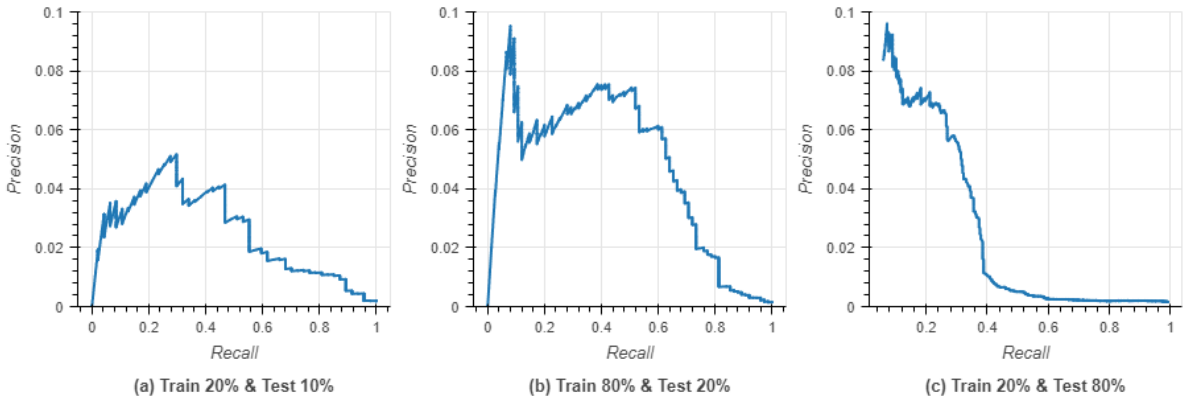


Figure 5.4: Precision-Recall Curve based on L_2 norm. (a) For 20% train and 10% test sets, (b) For 80% train and 20% test data sets, (c) For 20% train and 80% test data sets.

Figure 5.4(a) indicates that for 20% train and 10% test data sets we will get precision around 1% to capture 80% recall. This means that we get a high false positive rate. On the other hand Figure 5.4(b) shows that to capture 80% recall, we will get precision around 2% while the first 80% considered as training and the rest 20% as test. Finally we can see that in Figure 5.4(c) the precision will be around 0.2% for 80% recall, while the first 20% considered as training and the rest 80% as test.

5.2.2 L_2 distance to zero

We have also compared the L_2 distances with the distances to zero, to see whether the fraud cases are distinguishable or not. The scatter plot of $L2$ distances to zero is shown in Figure 5.5 for three different portions of training and test data sets.

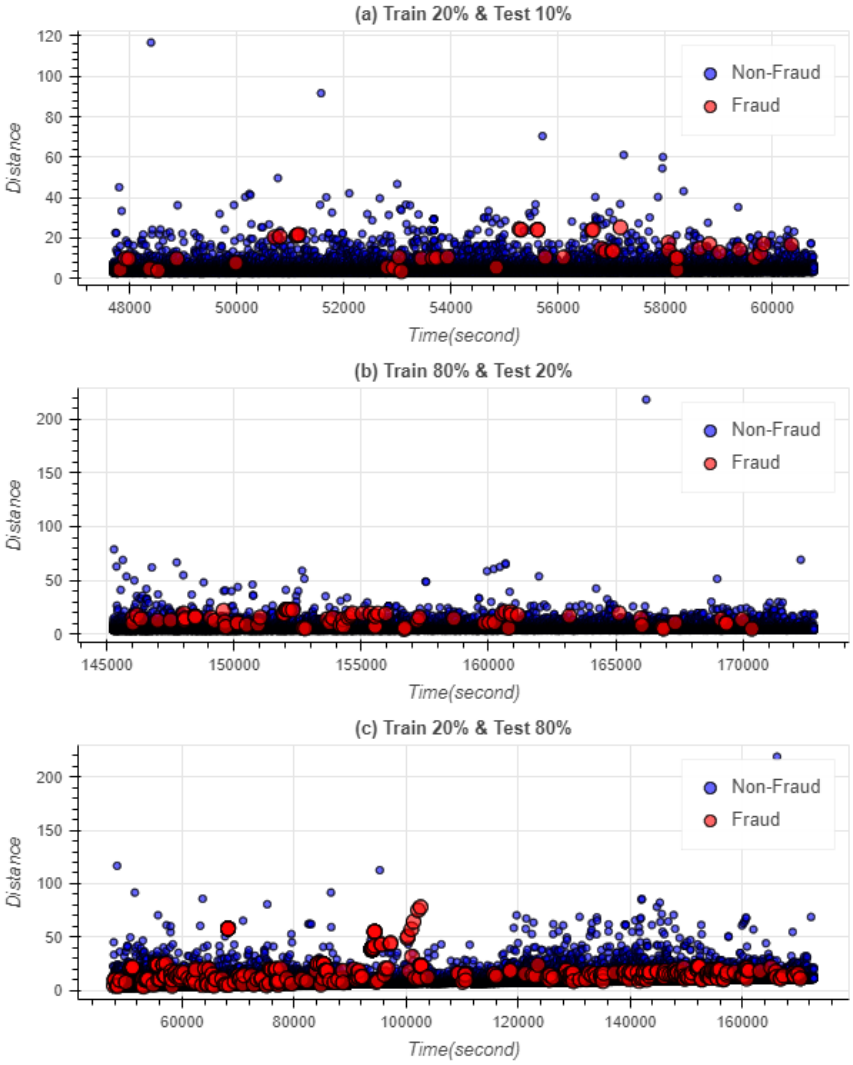


Figure 5.5: Scatter plot of distances based on L_2 distance to zero. (a) For 20% train and 10% test sets, (b) For 80% train and 20% test data sets, (c) For 20% train and 80% test data sets.

By comparing the L_2 distances with the distances to zero shown, it can be seen the fraud and non-fraud cases are not separating very well.

The performance of the L_2 distances to zero is checked by Precision-Recall Curve. We set the cut-off point between 10% and 99.9% quantiles of distance to zero, and computed the precision and recall values. The Precision-Recall Curve based on L_2 distance to zero is shown in Figure 5.6.

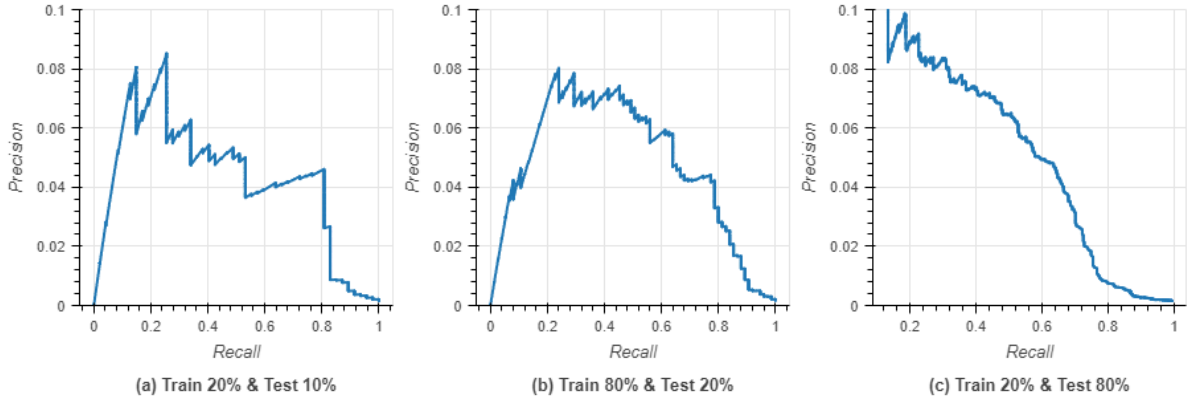


Figure 5.6: Precision-Recall Curve based on L_2 distance to zero. (a) For 20% train and 10% test sets, (b) For 80% train and 20% test data sets, (c) For 20% train and 80% test data sets.

It can be seen in Figure 5.6(a) that for 20% train and 10% test data sets we will get precision around 5% to capture 80% recall. On the other hand Figure 5.6(b) shows that to capture 80% recall, we will get precision around 3% while the first 80% considered as training and the rest 20% as test. Finally we can see that in Figure 5.6(c) the precision will be around 1% for 80% recall, while the first 20% considered as training and the rest 80% as test.

Additionally, we used the inverse L_2 distance for fraud detection in unsupervised method. The results are presented in Appendix A.1. Since there are several similar transactions done very closely over the time, the inverse L_2 distance values are very large in the beginning due to dependencies over the transaction time of the test data sets. Therefore, the inverse distance plots looked almost the same for different cut-offs.

5.2.3 Combination of L_2 distance and L_2 distance to zero

Another assumption that we have considered to predict the fraud and non-fraud cases, is to combine the L_2 distance and L_2 distance to zero suggested in Section 4.7. Since the distribution of L_2 distance and L_2 distance to zero are different in all cases, one can see that for combination with 'AND' condition, it's a small distance which dominates and for combination with 'OR' condition, it's a big distance which dominates.

In order to do more reliable comparison, we considered two cases in our analyses. First we standardized L_2 distance and L_2 distance to zero, and then we used uniform transformation of them. Then, we combined the cut-off points between 10% and 99.9% quantiles of each distances and defined a common range for cut-off points using the Equations 4.1 and 4.2 for fraud prediction.

Figures 5.7, 5.8 and 5.9 show the Precision-Recall curve for the combination of standardized L_2 distance and L_2 distance to zero for 20%/10%, 80%/20% and 20%/80%

for training and test sets splits respectively. In each figure, part (a) represents the 'OR' condition when at least one of the L_2 distance and L_2 distance to zero is greater than the cut-off point, and part (b) indicates the Precision-Recall curve for 'AND' condition when both distances are greater than the cut-off point.

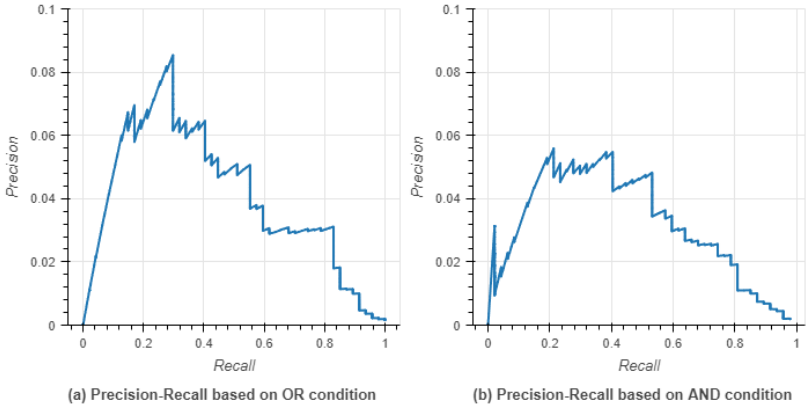


Figure 5.7: Precision-Recall Curve based on combination of standardized L_2 distance and L_2 distance to zero, while the first 20% considered as training and the next 10% as test. (a) PR curve for 'OR' condition. (b) PR curve for 'AND' condition.

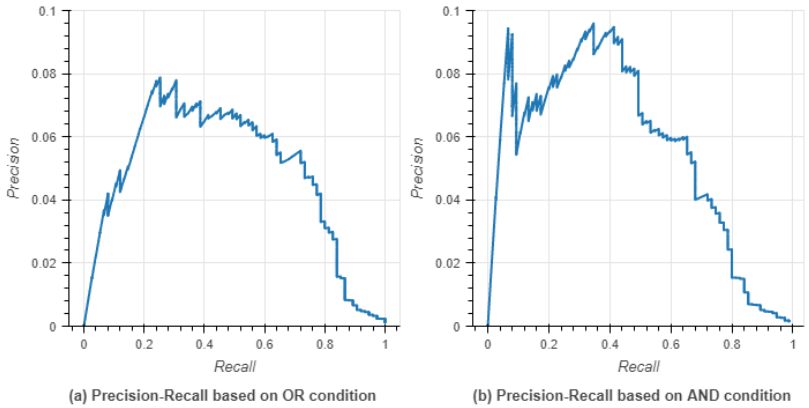


Figure 5.8: Precision-Recall Curve based on combination of standardized L_2 distance and L_2 distance to zero, while the first 80% considered as training and the next 20% as test. (a) PR curve for 'OR' condition. (b) PR curve for 'AND' condition.

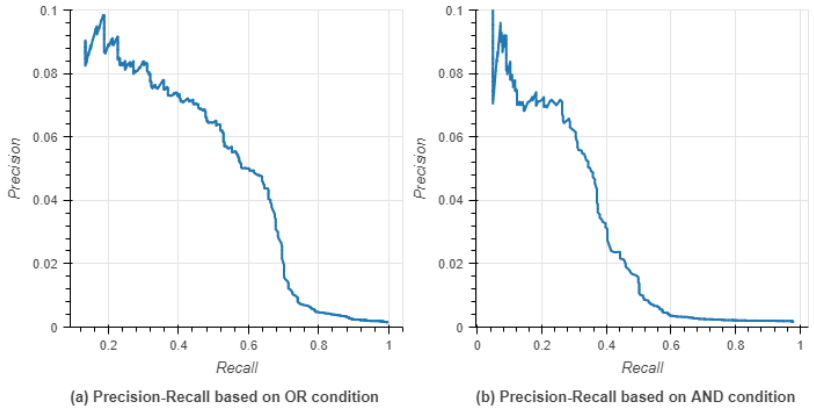


Figure 5.9: Precision-Recall Curve based on combination of standardized L_2 distance and L_2 distance to zero, while the first 20% considered as training and the next 80% as test. (a) PR curve for 'OR' condition. (b) PR curve for 'AND' condition.

In the three splits of 20/10, 80/20 and 20/80, the precision values at 80% recall for 'OR' condition are 3%, 3% and 0.2% and for 'AND' condition are 1%, 2% and 0.1% respectively. By comparing the area under the PR curves shown in Figures 5.7, 5.8 and 5.9, and considering the precision values, it can be seen that the split of 80/20 is performing better than the others with 'OR' condition.

In the next part, the combination of uniform transformation of L_2 distance and L_2 distance to zero for the three data splits are presented. Figures 5.10, 5.11 and 5.12 show the Precision-Recall curve for this combination. In each figure, part (a) represents the 'OR' condition when at least one of the L_2 distance and L_2 distance to zero is greater than the cut-off point, and part (b) indicates the Precision-Recall curve for 'AND' condition when both distances are greater than the cut-off point.

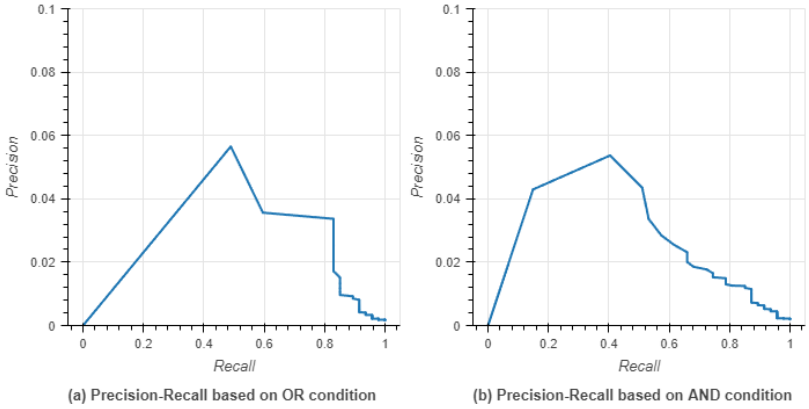


Figure 5.10: Precision-Recall Curve based on combination of uniform transformation of L_2 distance and L_2 distance to zero, while the first 20% considered as training and the next 10% as test. (a) PR curve for 'OR' condition. (b) PR curve for 'AND' condition.

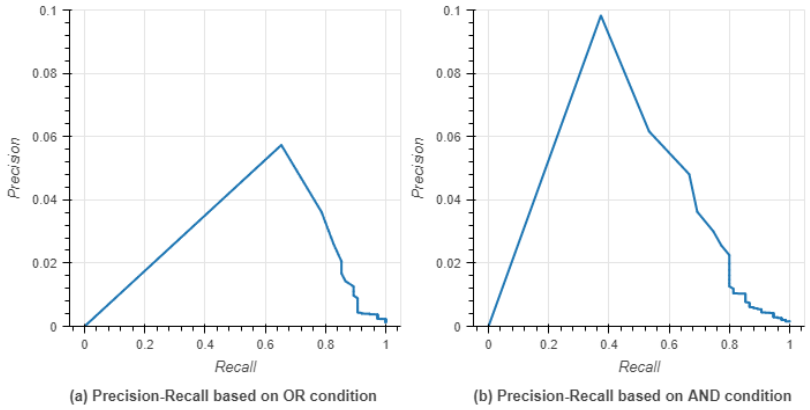


Figure 5.11: Precision-Recall Curve based on combination of uniform transformation of L_2 distance and L_2 distance to zero, while the first 80% considered as training and the next 20% as test. (a) PR curve for 'OR' condition. (b) PR curve for 'AND' condition.

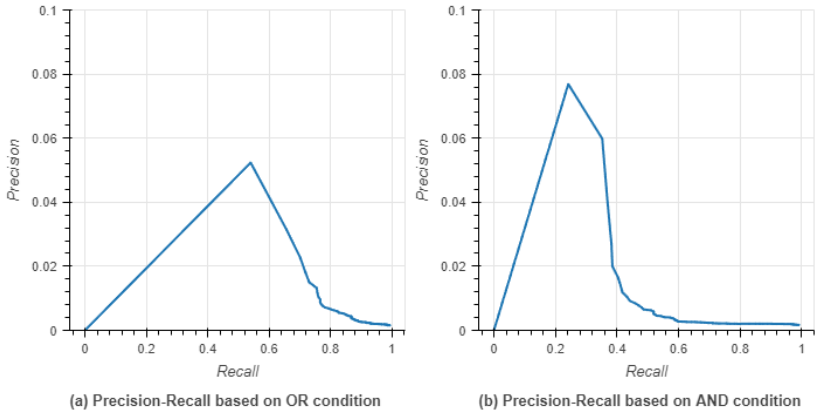


Figure 5.12: Precision-Recall Curve based on combination of uniform transformation of L_2 distance and L_2 distance to zero, while the first 20% considered as training and the next 80% as test. (a) PR curve for 'OR' condition. (b) PR curve for 'AND' condition.

By looking at parts (a) and (b) for the three splits in Figures 5.10, 5.11 and 5.12, one can see that 80/20 split is typically performing better that the other splits as the area under the Precision-Recall curve is larger than the two other cases.

The results from the combination of distances for the three data splits without considering any standardization and uniform transformation are presented in A.2. It can be observed that in a combination of distances with 'AND' condition, it's a small distance which dominates and in a combination with 'OR' condition, it's a big distance which dominates.

5.2.4 Summary

In the unsupervised method, the analysis has been done on the three data splits of 20/10, 80/20 and 20/80 for training and test sets. The distances used in unsupervised method are L_2 distance, L_2 distance to zero and inverse L_2 distance.

To predict whether a point from the test set is fraud or not, we looked at the distance of its nearest neighbor from the training set and assigned it as a fraud case if the distance is larger than the predefined threshold. We assumed that the fraud cases have the highest distances on average. However, the scatter plots of the three distances illustrated in Figures 5.3, 5.5 and A.1, showed that the fraud cases are located and mixed in the middle of the normal cases for all three portions and it is not easy to distinguish between fraud and non-fraud cases.

The performance of unsupervised methods were evaluated by the precision-recall curves. It is more important to have high recall than having high precision, and in reality we might only want to miss 20 percent of the real fraud which means that we have to have around 80 percent recall. By looking at different Precision-Recall curves, we can see that the precision will be nearly 1 to 5 percent. Therefore, we will get a lot of false positives that should be checked to be able to not missing any fraud cases. In fact for the fraud detection, the 80 to 90 percent is the most important part of Precision-Recall plots.

Based on both area under curve and the precision values at 80% recall for the L_2 distance, the L_2 distance to zero, and the standardized "OR" combinations are similar, but with L_2 distance to zero slightly better. Still one might want to use standardized "OR" to be on the safe side. Combinations using transformation to uniformity performed a little worse, using combinations without transforming to the same scale just copied L_2 distance and L_2 distance to zero, respectively, and inverse distance didn't work at all.

6

Conclusion

In this chapter, we mention our general conclusion about the supervised and unsupervised nearest neighbor distance methods. The results from the performance comparison of the methods and the examined approaches are discussed. Finally we provide some suggestions about possible future research.

6.1 General Overview

As a financial investigator, having access to information about the time of detecting fraud situations is matter and it might be better to take the already available small portion of labeled data for training and do the prediction for the upcoming transactions. On the other hand, in some cases it might be better to wait until getting the large portion of labeled data to be ready to be able to do the investigation for the rest of the data. In particular, the main reason for dividing the data set into the different portion of train and test sets correspond to different practical situations. It depends also on what kind of the data one has access to. Even though someone has access to for example 80% of data, maybe it is better to use only 20% of the data as the time latency for computation of the analysis in 80% is increasing.

In a fraud situation, it is more important to not miss the fraud cases. So we need to have more recall than precision based on the algorithms. This means that we will get more false positive which one need to check them. Of course in the real situations, it depends on the capacity of a company for checking these many non-fraud cases. For example, in banks, they might prefer to not miss any fraud cases and will check all those non-fraud cases which detected as fraud to make sure that they won't put their costumers at risk.

The supervised methods are quite efficient but require the availability of labeled training data i.e. one knows which transactions are frauds. However in real world, labeled data are often not available and then one has to use unsupervised based method. In other words, even though supervised learning is better, when there is no access to labeled data, then one has to use unsupervised methods.

6.2 Discussion

In this thesis, we have conducted an investigation on methods for the detection of fraudulent cases in credit card transactions. We have used nearest neighbor algorithms based on supervised and unsupervised methods. In the supervised method, we looked at the label of closest neighbor of each point from test data set and used it as the predicted label for the test point. In the unsupervised version, for each data point from the test set we looked at their distances to the closest point from the training set and assigned them as a fraud case if the distances are larger than the established threshold. The performance of L_2 distance, L_2 distance to zero and inverse L_2 distance methods were evaluated by means of precision-recall curves.

In the supervised nearest neighbor method, we made a model that find the first nearest neighbor, the first two, the first three, and the first four nearest neighbors in order to predict the label of the test data points. We assigned label fraud to a test point if at least one of its nearest neighbors has label of fraud. In the unsupervised version of nearest neighbor distance algorithm, the general thinking is that the fraud cases should be far from the others or be isolated in an area. However in our analysis, we can see the distance based algorithm couldn't distinguish between the fraud and non-fraud cases.

The results from the tables 5.2 (a-c), showed that if k as the number of neighbors is getting larger, the false negative errors go down and the false positive errors go up. Based on the precision and recall values, it is highly important to have a high recall value than having high precision as a financial investigator since one doesn't want to miss fraud cases and minimizes the false negatives and maximize the true positives.

Based on both area under curve and precision at 80% the L_2 distance, the L_2 distance to zero, and the standardized 'OR' combinations are similar, but with L_2 distance to zero slightly better. To be more safe and not lose any information, one could still want to use the standardized combination method with 'OR' condition. Combinations using transformation to uniformity performed a little worse, using combinations without transforming to the same scale just copied L_2 distance and L_2 distance to zero, respectively, and inverse distance didn't work at all.

If the PR curve in one method is completely over the PR curve of another method then the curve which is located on the top is the best one. If the curves are crossed then maybe one curve is better to the left and less good to the right. In credit card fraud it is often the right part of the curve is important because it's typically worse to miss frauds. In general, a financial investigator doesn't want to lose fraud cases and it takes cost doing more checking to identify fraud transactions.

6.3 Future Research

The papers [15] and [22] introduce an approach called the *reverse* nearest neighbor outlier detection method. In this approach, one looks at each point in a data set and counts how many times that point was the neighbour of other points in the data set. The idea is that if a point turns out rarely as a neighbor of other points, it will may be an outlier. As future research, it would be interesting to investigate how reverse nearest neighbor approach works for detection of fraud in credit card transactions.

Bibliography

[1] Financial Fraud, Investopedia, https://www.investopedia.com/financial-fraud-4689710

[2] J. West, M. Bhattacharya, R. Islam, Intelligent Financial Fraud Detection Practices: An Investigation, School of Computing & Mathematics, Charles Sturt University (2015)

[3] C. Mohan, K. Mehrotra, Anomaly detection in banking operations, (2017), https://www.semanticscholar.org/paper/Anomaly-detection-in-banking-operations-Mohan-Mehrotra/b48afd871d0f2d846a9d1d8ab682bc70da12dde3

[4] A. Banarescho, Detecting and Preventing Fraud with Data Analytics, Procedia Economics and Finance, Vol. 32 (2015) pp. 1827-1836.

[5] M. Vasarhelyi, H. Issa, Application of Anomaly Detection Techniques to Identify Fraudulent Refunds, (2011) https://dx.doi.org/10.2139/ssrn.1910468

[6] M. Ahmed, A. Mahmood, M. Islam, A survey of anomaly detection techniques in financial domain, Future Generation Computer Systems, Vol. 55 (2016) pp. 278-288.

[7] A. Patcha, J. M. Park, An overview of anomaly detection techniques: Existing solutions and latest technological trends, Computer Networks (2007), pp. 3448-3470.

[8] A. Sabau, Survey of Clustering based Financial Fraud Detection Research, Informatica Economica Journal, Vol. 16 (2012), pp. 110-122.

[9] Sh. Agrawal, J. Agrawal, Survey on Anomaly Detection using Data Mining Techniques, Procedia Computer Science; Vol. 60 (2015), pp. 708-713.

[10] W. Hilal, S.A. Gadsden, J. Yawney, Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances, Expert Systems With Applications (2022)

[11] John Ulzheimer. How Do Credit Card Issuers Detect Fraud? - Credit Card Insider. YouTube. 2015. URL: https://www.youtube.com/watch?v=N3LjCsVYQ-k

[12] Kaggle, Credit Card Fraud Detection, https://www.kaggle.com/mlg-ulb/creditcardfraud

[13] K. Muameleci, Anomaly Detection in Credit Card Transactions using Multivariate Generalized Pareto Distribution, Chalmers University of Technology (2022) https://odr.chalmers.se/handle/20.500.12380/304780

[14] S. Thudumu, P. Branch, J. Jin, J. Singh, A comprehensive survey of anomaly detection techniques for high dimensional big data, Journal of Big Data (2020).

[15] M. Radovanović, A. Nanopoulos, M. Ivanović, Reverse Nearest Neighbors in Unsupervised Distance-Based Outlier Detection, IEEE Transactions on Knowledge and Data Engineering (May 2015)

[16] H. Bodepudi, Credit Card Fraud Detection Using Unsupervised Machine Learning Algorithms, International Journal of Computer Trends and Technology (2021) https://ijcttjournal.org/archives/ijctt-v69i8p101

[17] K. Vishwakarma, Credit Card Fraudulent Detection using Machine Learning, International Research Journal of Engineering and Technology (IRJET). Vol. 7, Issue: 09, (Sep 2020).

[18] M. Ummul Safa, R. M. Ganga, Credit Card Fraud Detection Using Machine Learning, International Journal of Research in Engineering, Science and Management. Vol. 2, Issue: 11, (Nov 2019).

[19] M. Zhao, J. Chen, Y. Li, A Review of Anomaly Detection Techniques Based on Nearest Neighbor (2018) https://www.atlantis-press.com/article/25897526.pdf

[20] C. Navamani, M. Phil, Credit Card Nearest Neighbor Based Outlier Detection Techniques, International Journal of Computer Techniques. Vol. 5, Issue: 2 (2018) http://www.ijctjournal.org/Volume5/Issue2/IJCT-V5I2P8.pdf

[21] J. S.Gosavi , V. S.Wadne, Unsupervised Distance-Based Outlier Detection Using Nearest Neighbours Algorithm on Distributed Approach: Survey (2014) https://www.semanticscholar.org/paper/Unsupervised-Distance-Based-Outlier-Detection-Using-Gosavi-Wadne/d414d89a1d13c4ae7a73ad274e3ea320289d6369

[22] Altman, N. S. 1992. An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician 46(3):175–185.

[23] C. Cortes, V. Vapnik, 1995. Support-vector networks. Machine learning 20(3):273–297.

[24] E. Ileberi, Y. Sun, Z. Wang, A machine learning based credit card fraud detection using the GA algorithm for feature selection, Journal of Big Data (2022) 9:24.

[25] Y. Sahin, E. Duman, (2011). Detecting Credit Card Fraud by Decision Trees and Support Vector Machines, International MultiConference of Engineers and Computer Scientists, Hong Kong.

[26] L. Breiman, Random forests, Machine Learning 45, 5 (2001).

[27] O. Johansson, Anomaly detection: Credit card fraud detection using Variational Autoencoder, One-class Classifier and Random Forest, Chalmers University of Technology, (2022)

[28] M. Gumbao, (2019, June 03). Best clustering algorithms for anomaly detection. Retrieved from Towards Data Science, available online: https://towardsdatascience.com/best-clustering-algorithms-for-anomaly-detection-d5b7412537c8

[29] T. Hastie, R. Tibshirani, Jerome Friedman, (2008) The Elements of Statistical Learning, Springer Series in Statistics

[30] M. H. Ahmed, Credit Card Fraud Detection Techniques: A Survey, University of Management and Technology (2022)

[31] X. Niu, Li Wang, X. Yang, A Comparison Study of Credit Card Fraud Detection: Supervised versus Unsupervised, (2019)

[32] M. Rezapour, Anomaly Detection using Unsupervised Methods: Credit Card Fraud Case Study, International Journal of Advanced Computer Science and Applications. Vol. 10, No. 11, (2019)

[33] V U. Mahesh, S. K. Reddy, L. Reddy, S. Krishna, J. Dilleshwar, S. Kaur, Credit Card Fraud Detection using Unsupervised Machine Learning, International Research Journal of Engineering and Technology (IRJET) Vol. 08 Issue: 04 (Apr 2021)

[34] Intuition- brute force, KD & Ball tree, learndataa, YouTube. 2021. URL: https://www.youtube.com/watch?v=FnJj28u7rF0

[35] H. Marius, (Jun 15, 2020). Tree algorithms explained: Ball Tree Algorithm vs. KD Tree vs. Brute Force. Retrieved from Towards Data Science: https://towardsdatascience.com/tree-algorithms-explained-ball-tree-algorithm-vs-kd-tree-vs-brute-force-9746debcd940

[36] Nearest Neighbors. Retrieved from Scikit Learn. URL: https://scikit-learn.org/stable/modules/neighbors.html

[37] Precision-Recall, Retrieved from Scikit Learn, URL: $https : //scikit - learn.org/stable/auto_examples/model_selection/plot_precision_recall.html$

[38] V. N. Dornadulaa , S. Geetha, Credit Card Fraud Detection using Machine Learning Algorithms, Procedia Computer Science 165 (2019) 631–641, https://doi.org/10.1016/j.procs.2020.01.057

[39] P. Tiwari, S. Mehta, N. Sakhuja, J. Kumar, and A. K. Singh, 2021. Credit card fraud detection using machine learning: A study, https://arxiv.org/abs/2108.10005

A

Appendix

In this Appendix, we provided more results based on unsupervised nearest neighbor algorithm with inverse L_2 distance. The results for combination of L_2 distance and L_2 distance to zero without considering any standardization and uniform transformation is also presented in this section. The algorithm applied to the three splits of 20/10, 80/20 and 20/80 for training and test sets.

A.1 Inverse L_2 distance

The nearest neighbor distance algorithm is used based on inverse L_2 distance for different portions of training and test sets. For each point from the test set, we found the closest point from the training set based on the minimum distance and took the inverse of distances. The scatter plot of inverse L_2 distances is shown in Figure A.1.

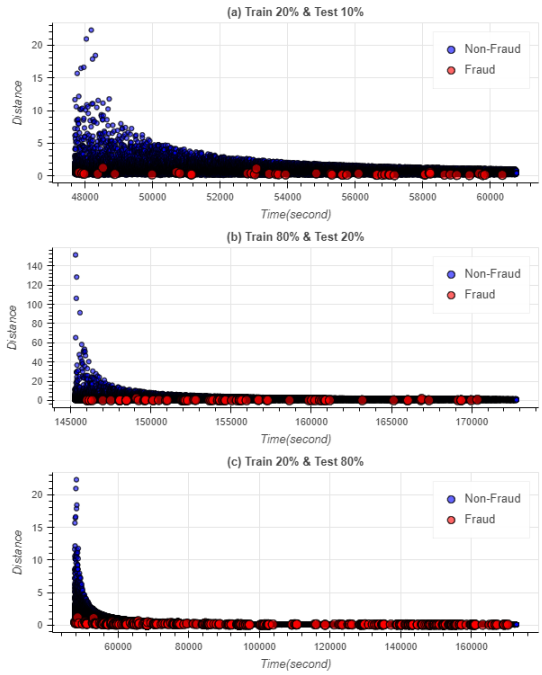


Figure A.1: Scatter plot of distances based on inverse L_2 norm.(a) for 20% train and 10% test sets. (b) for 80% train and 20% test data sets. (c) for 20% train and 80% test data sets.

The performance of the inverse L_2 distances is checked by Precision-Recall Curve computed for different cut-off points between 10% and 99.9% quantiles of inverse L_2 distances. Figure A.2 shows the Precision-Recall Curve based on inverse L_2 distances.

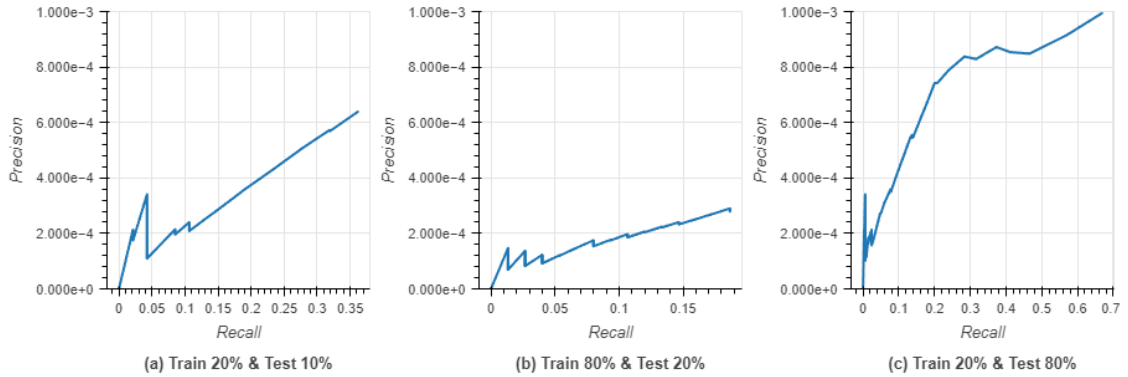


Figure A.2: Precision-Recall Curve based on inverse L_2 distance. (a) For 20% train and 10% test sets, (b) For 80% train and 20% test data sets, (c) For 20% train and 80% test data sets.

It can be seen in Figures A.2 (a)-(c) that the recall values in all the three curves don't go out all the way until recalls higher as high as 80% or higher.

In general, there are several very similar transactions done very closely over the time and due to the dependencies the inverse L_2 distance values are very large in the beginning over the test time. Therefore, the inverse distance plots looks almost the same for different cut-off values.

A.2 Combination of L_2 distance and L_2 distance to zero without any transformation

As mentioned in Section 4.7, another way that have been considered for predicting the fraud and non-fraud cases was to combine the L_2 distance and L_2 distance to zero. Here, the results are presented based on non-transformed L_2 distance and L_2 distance to zero. We combined the cut-off points between 10% and 99.9% quantiles of each distances and defined a common range for cut-off points to predict the fraud cases using the Equations 4.1 and 4.2. Figures A.3, A.4 and A.5 show the Precision-Recall curves for three different portions of training and test sets. In each figure, part (a) shows the curve when at least one of the L_2 distance and L_2 distance to zero is greater than the cut-off point, and part (b) illustrates the curve when both distances are greater than the cut-off point.

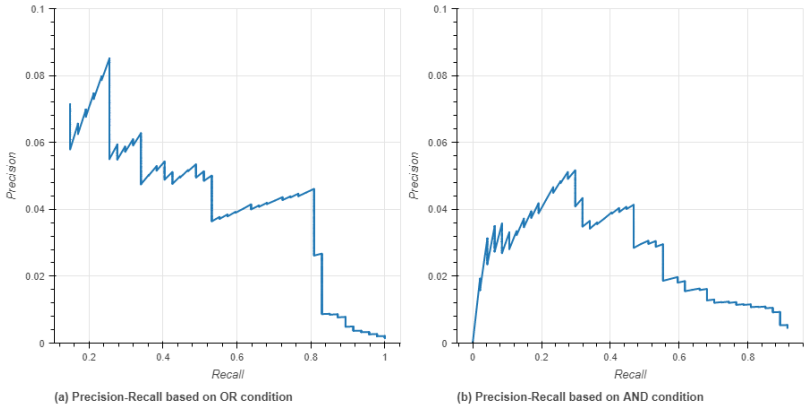


Figure A.3: Precision-Recall Curve based on the combination of L_2 distance and L_2 distance to zero, while the first 20% considered as training and the next 10% as test. (a) PR curve for 'OR' condition. (b) PR curve for 'AND' condition.

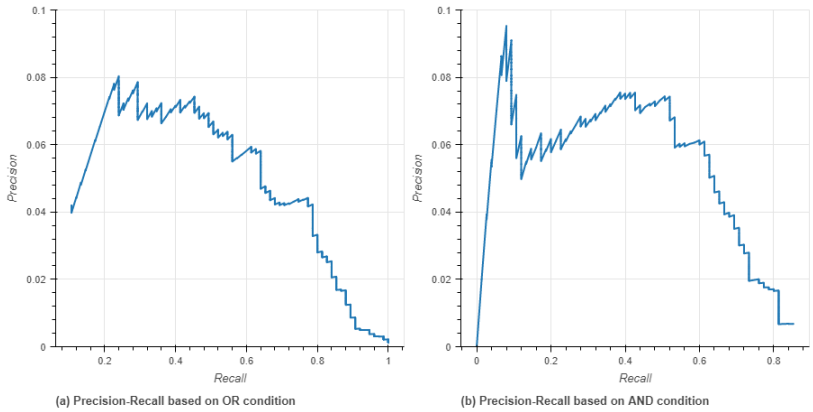


Figure A.4: Precision-Recall Curve based on the combination of L_2 distance and L_2 distance to zero, while the first 80% considered as training and the next 20% as test. (a) PR curve for 'OR' condition. (b) PR curve for 'AND' condition.

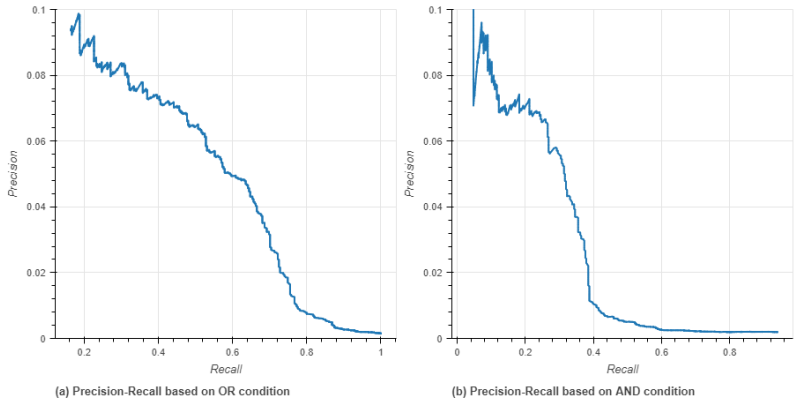


Figure A.5: Precision-Recall Curve based on the combination of L_2 distance and L_2 distance to zero, while the first 20% considered as training and the next 80% as test. (a) PR curve for 'OR' condition. (b) PR curve for 'AND' condition.

It can be seen that in all figures, the curves are shown in part (a) are similar with the corresponding L_2 distance to zero curves when we have OR condition, and the curves in part (b) are almost the same as the corresponding L_2 distance when we set AND condition for combining both distances. The reason for that is, the distribution of L_2 distance and L_2 distance to zero are different in all cases. If we have AND condition, it's a small distance which dominates and if we have OR condition, then it's a big distance which dominates.