Decision Support System to Predict Credit Card Fraud

Business Intelligence for Decision Support Spring 2018

Abhishek Dangi (12736625)
Lohit Ravindra (12795389)
Richa Vyas (12640304)

Table of Contents

Abstract

Data mining is becoming strategically important area for many business organizations including banking sector. It is a process of analysing the data from various perspectives and summarizing it into valuable information. Data mining assists the banks to look for hidden pattern in a group and discover unknown relationship in the data. These techniques facilitate useful data interpretations for the banking sector. Fraud is a significant problem in banking sector. Detecting and preventing fraud is difficult, because fraudsters develop new schemes all the time, and the schemes grow more and more sophisticated to elude easy detection. This report analyses the data mining techniques and its applications in banking sector initially from a general point of view. Then we present an in-depth analysis of one of the major frauds the banks face – Credit Card fraud. The literature review presents a survey of various techniques which are used in credit card fraud detection mechanisms. Then all of those techniques are analysed in terms of Cost, Accuracy and Speed of Detection. To further our analysis, we have presented a case study to explain the in-depth processes involved in Credit Card Fraud detection. We have added a special section to understand how accurately the decision is taken by this system.

Keywords

Banking Sector, Customer Retention, Data mining, Fraud Detection, Credit card fraud, credit card fraud detection methods, E-commerce

1 Introduction:

1.1 Data Mining Techniques and its Applications in Banking Sector

Technological innovations have enabled the banking industry to open up efficient delivery channels. IT has helped the banking industry to deal with the challenges the new economy poses. Nowadays, Banks have realized that customer relationships are a very important factor for their success. Customer relationship management (CRM) is a strategy that can help them to build long-lasting relationships with their customers and increase their revenues and profits. CRM in the banking sector is of greater importance. The CRM focus is shifting from customer acquisition to customer retention and ensuring the appropriate amounts of time, money and managerial resources are directed at both of these key tasks. The challenge the bank face is how to retain the most profitable customers and how to do that at the lowest cost. At the same time, they need to find and implement this solution quickly and the solution to be flexible. Traditional methods of data analysis have long been used to detect fraud. They require complex and time-consuming investigations that deal with different domains of knowledge like financial, economics, business practices and law. Fraud instances can be similar in content and appearance but usually are not identical. In developing countries like India, Bankers face more problems with the fraudsters. Using data mining technique, it is simple to build a successful predictive model and visualize the report into meaningful information to the user.

Data mining tools, using large databases, can facilitate

1. Automatic prediction of future trends and behaviours and
2. Automated discovery of previously unknown patterns

1.2 Data Mining Techniques and Algorithms

Data mining algorithms specify a variety of problems that can be modelled and solved. Data mining functions fall generally into two categories:

1. Supervised Learning
2. Unsupervised Learning

Concepts of supervised and unsupervised learning are derived from the science of machine learning, which has been called a sub-area of artificial intelligence. Artificial intelligence means the implementation and study of systems that exhibit autonomous intelligence or behaviour of their own. Machine learning deals with techniques that enable devices to learn from their own performance and modify their own functioning. Data mining applies machine learning concepts to data.

1.2.1 Supervised Learning

Supervised learning is also known as directed learning. The learning process is directed by a previously known dependent attribute or target. Directed data mining attempts to explain the behaviour of the target as a function of a set of independent attributes or predictors. Supervised learning generally results in predictive models. This is in contrast to unsupervised learning

where the goal is pattern detection. The building of a supervised model involves training, a process whereby the software analyses many cases where the target value is already known. In the training process, the model "learns" the logic for making the prediction. For example, a model that seeks to identify the customers who are likely to respond to a promotion must be trained by analysing the characteristics of many customers who are known to have responded or not responded to a promotion in the past.

Table 1: Algorithms preferred for Supervised learning with description

Algorithm	Function	Description
Decision Tree	Classification	Decision trees extract predictive information in the form of human-understandable rules. The rules are if-then-else expressions; they explain the decisions that lead to the prediction.
Generalized Linear Models (GLM)	Classification and Regression	GLM implements logistic regression for classification of binary targets and linear regression for continuous targets. GLM classification supports confidence bounds for prediction probabilities. GLM regression supports confidence bounds for predictions.
Minimum Description Length (MDL)	Attribute Importance	MDL is an information theoretic model selection principle. MDL assumes that the simplest, most compact representation of data is the best and most probable explanation of the data.
Naive Bayes (NB)	Classification	Naive Bayes makes predictions using Bayes' Theorem, which derives the probability of a prediction from the underlying evidence, as observed in the data.
Support Vector Machine (SVM)	Classification and Regression	Distinct versions of SVM use different kernel functions to handle different types of data sets. Linear and Gaussian (nonlinear) kernels are supported. SVM classification attempts to separate the target classes with the widest possible margin. SVM regression tries to find a continuous function such that the maximum number of data points lie within an epsilon-wide tube around it.

1.2.2 Unsupervised Learning

Unsupervised learning is non-directed. There is no distinction between dependent and independent attributes. There is no previously-known result to guide the algorithm in building the model. Unsupervised learning can be used for descriptive purposes. It can also be used to make predictions.

Table 2: Algorithms preferred for Unsupervised learning with description

Algorithm	Function	Description
Apriori	Association	Apriori performs market basket analysis by discovering co-occurring items (frequent itemsets) within a set. Apriori finds rules with support greater than a specified minimum support and confidence greater than a specified minimum confidence. For example Find the items that tend to be purchased together and specify their relationship
k-Means	Clustering	k-Means is a distance-based clustering algorithm that partitions the data into a predetermined number of clusters. Each cluster has a centroid (center of gravity). Cases (individuals within the population) that are in a cluster are close to the centroid. For example, segment demographic data into clusters and rank the probability that an individual will belong to a given cluster
Non-Negative Matrix Factorization (NMF)	Feature Extraction	NMF generates new attributes using linear combinations of the original attributes. The coefficients of the linear combinations are non-negative. During model apply, an NMF model maps the original data into the new set of attributes discovered by the model. For example, given demographic data about a set of customers, group the attributes into general characteristics of the customers
One Class Support Vector Machine	Anomaly Detection	One-class SVM builds a profile of one class and when applied, flags cases that are somehow different from that profile. This allows for the detection of rare cases that are not necessarily related to each other. For example, given demographic data about a set of customers, identify customer transaction behavior that is significantly different from the normal.

1.3 Top 10 Frauds in Banking Sector

The Reserve Bank of India – RBI maintains data on frauds on the basis of area of operation under which the frauds have been perpetrated. According to such data pertaining, top 10 categories under which frauds have been reported by banks are as follows

1. Credit Cards
2. Deposits – Savings A/C
3. Internet Banking
4. Housing Loans
5. Term Loans
6. Cheque / Demand Drafts
7. Cash Transactions
8. Cash Credit A/c (Types of Overdraft A/C]
9. Advances
10. ATM / Debit Cards

For the purpose of this report we will be focussing on Credit Card Frauds.

As there is a vast advancement in the E-commerce technology, the use of credit cards has reached an all-time high. Credit cards have become a crucial mode of payment. With the rise in the credit card transactions, the credit card frauds have also become frequent nowadays. Thus, an improved fraud detection system has become essential to maintain the reliability of the payment system. The criterion is to assure secured transactions for credit card owners so that they can make electronic payment safely for the services and goods which are provided on internet. In an e-bank many transactions undergo simultaneously, so a fraud detection system should distinguish between legitimate, suspicious fraud and an illegitimate transaction. There are many modern and new techniques which are based on Neural Network, Artificial Intelligence, Bayesian Network, Data mining, Artificial Immune System, K- nearest neighbour algorithm, Decision Tree, Fuzzy Logic Based System, Support Vector Machine, Machine learning, Genetic Programming etc., that has developed in detecting various credit card fraudulent transactions. This report represents a survey of various techniques which are used in credit card fraud detection mechanisms.

2.Literature Review

2.1 Card Fraud Detection Techniques

E-commerce payment systems have become increasingly popular due to the widespread use of the internet-based shopping and banking. Credit Card Fraud is one of the largest threats to business organizations today. However, to overpower the fraud effectively, it is important to first understand the mechanisms of executing a fraud i.e. we need to understand the techniques of cyber credit card frauds. Since earlier the fraud is detected only when the billing for credit card is done, it is very hard to prevent fraudulent transactions. Therefore, the need to assure unexposed transactions for credit-card owners when using their credit cards to make electronic payments for goods and services provided on the internet is a criterion.

2.2 Types of Frauds

The credit-card fraud is divided into two types;

1. The online credit card fraud (or no card present fraud) and
2. The offline credit card fraud (card present fraud)

The online credit-card fraud (also known as cyber credit card fraud) is committed with no presence of a credit-card but instead, the use of a credit-card information to make electronic purchase for goods and services on the internet. The offline credit-card fraud is committed with the presence of a credit-card which in most cases have been stolen or fake and thereby used at a local store or a physical location for the purchase or some goods or services.

There are many cyber credit card fraudsters. Some of them are:

(i) **Credit-card information buyers:** these are the fraudsters who either have little or no professional computer skills like computer programming, networking etc. They buy stolen or hacked credit card information on an illegal credit card sales website, with the intension of buying goods and products online.

(ii) **Physical credit-card stealers:** these are the fraudsters who physically steal credit cards may be by pick pocketing and use the information on it for making e-payment on internet for shopping.

(iii) **Black hat hackers:** "Black hat hackers" which are also known as a cracker are those who violate computer security with malevolent intentions or for personal gains. They choose their targets using a two-pronged process known as the "pre-hacking stage"; which includes Targeting, Research and Information Gathering and finally finishing the Attack. These hackers are highly skilled in Computer Programming and Computer Networking and with such skills they can barge in a network of computers. The main purpose of their act of intrusion or hacking is to steal personal or private information such as credit-card information, bank-account information, etc. for their own personal gains.

2.3 Techniques Used by Credit Card Fraudsters

In order to detect cyber credit-card fraud activities on the internet, a study was conducted on how credit-card information is stolen. Here are some of the different techniques which are used for credit-card fraud information theft.

(i) Credit-card fraud generator software: This software is used to generate valid credit-card numbers and expiry dates. Some of these software's are capable in generating valid credit-card numbers like credit-card companies or issuers because it uses the mathematical Luhn algorithm that credit-card companies or issuers use in generating credit-card numbers to their credit-card consumers or users. In some cases, this software is written by black-hat hackers who have hacked credit-card information stored on a database file from which the software can display valid credit-card information to other type of cyber credit-card fraudsters who have

bought the software to use. This technique in some cases is used by black-hat hackers to sell their hacked credit-card information to other online credit-card fraudsters with little or no computer skills.

(ii) Key –logger and Sniffers: The Black-hat hackers who have professional Programming or computer skills infect a computer by installing and automatically running sniffers or key-logger computer programs by which, they log all the keyboard inputs made into the computer on a file with the intention of retrieving personal information like credit-card information, etc. These fraudsters are able to infect the user's computers by sending infectious spam mails to computer users & asking them to download free games or software, and when those are downloaded, the sniffers of key-loggers are downloaded automatically, installed and ran on the user's computers. While the sniffer is running under the user's computer, they ken and log all the keyboard inputs made by the user over a network. Therefore, any user can unknowingly share their private information through this infectious software. Sometimes this software is also shared or sold to other fraudsters who do not have computer knowledge or skills.

(iii) Site-cloning, Spyware and Merchant sites: This software is also created by black-hat hackers, which are installed and ran on user's computer to keep track of all the website activities. By tracking and knowing the website activities of the user on the internet, they clone the electronic or banking websites which are regularly visited by the user and send the user for using it with the intension of retrieving private or personal information. Also, in the case of fake merchant sites, fake websites are created on which cheap products are provided to users and thereby asking user for payment by credit cards. If any payment is made on these fake sites, the user's credit card information is then stolen.

(iv) Physically stolen credit-card information: The fraudsters can steal the credit card and use the information to buying goods and products online.

(v) CC/CVV2 shopping websites: cyber credit-card fraudsters who have no professional computer skills buy hacked credit-card information on these websites to use for fraudulent electronic payment for some goods and services on the internet.

2.4 Credit Card Fraud Detection Methods

On doing the literature survey of various methods for fraud detection we conclude that to detect credit card fraud there are a lot of approaches.

- A Hybrid Approach Using Dempster-Shafer Theory and Bayesian Theory.
- Blast-SSAHA Hybridization
- Hidden Markov Model
- Genetic Algorithm
- Neural Network
- Bayesian Network
- K- Nearest Neighbour algorithm
- Stream Outlier Detection based on Reverse K-Nearest Neighbours (SODRNN)

- Fuzzy Logic Based System
- Decision Tree
- Fuzzy Expert System
- Support Vector Machine
- Meta Learning Strategy

2.4.1 A Hybrid Approach Using Dempster-Shafer Theory and Bayesian Theory

This Credit card fraud detection system is based on the integration of three approaches, i.e., rule-based filtering, Dempster– Shafer theory and Bayesian learning. Dempster's rule is applied to combine and associate multiple evidences from the rule- based component for calculation of initial belief about every incoming transaction. The suspicion score is updated through Bayesian learning using history database of both genuine cardholder as well as fraudster. THD is the transaction repository component of the above fraud detection system. History records of both fraudulent as well as genuine transactions are used to construct systems which allow us to extract characteristic information of the two groups from available data. For performing this, a good transaction history (GTH) for individual customers from their past behaviour and a generic fraud transaction history (FTH) from different types of past fraud data is build.

Each history transaction is represented by a set of attributes which contains information like card number, transaction amount and time since the last purchase was made. While observing the current spending behaviour on a credit card, the past spending behaviour in terms of the frequency of transactions on that card are also accumulated and analysed. The transaction amount information in the THD is needed for detecting the outliers. The FDS architecture is flexible so that new rules using any other effective technique can be included at a later stage to further grow the rule-based component. Bayesian learning contributes to the FDS by helping it to dynamically adapt to the changing behaviour of genuine customers as well as fraudsters over time. The Dempster–Shafer theory gives good performance, especially in terms of true positives, and Bayesian learning helps to further improve the system accuracy.
It has high accuracy. It reduces false alarms and improves detection rate and also applicable in E-Commerce. But it is very expensive, and its processing Speed is also low.

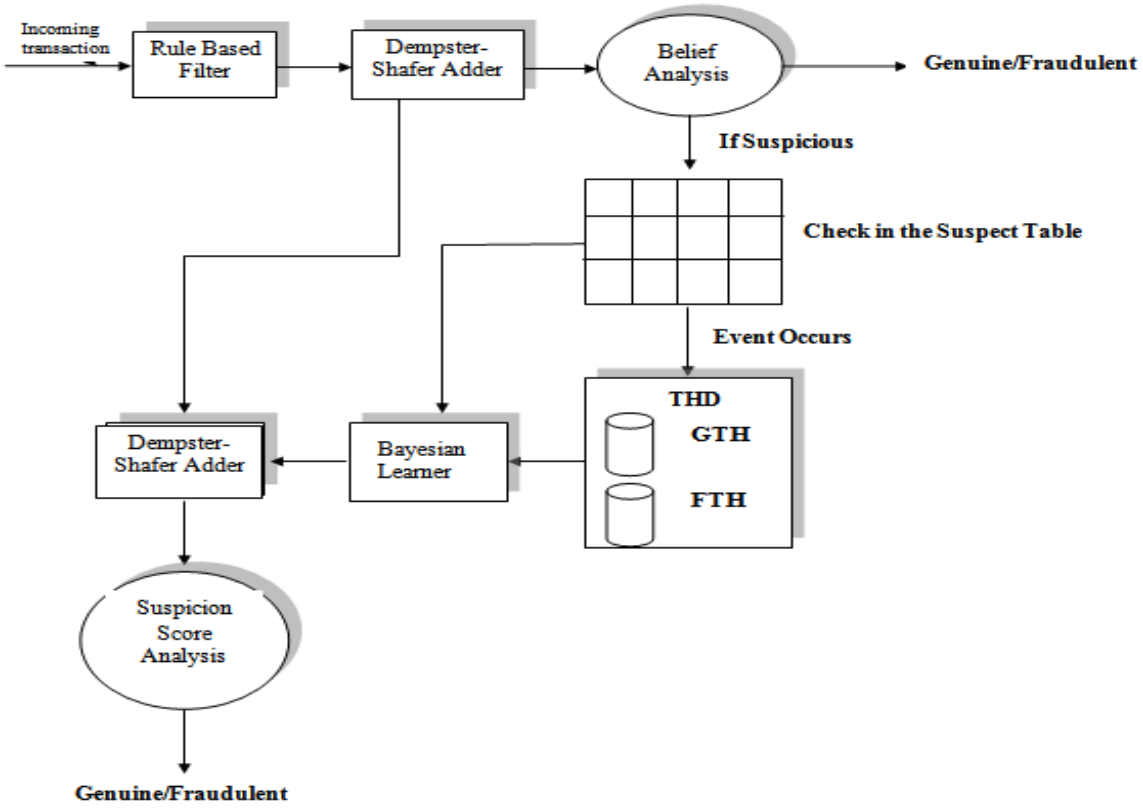


Figure 1: Block Diagram of Fraud Detection System Using Dempster-Shafer Theory and Bayesian Network

2.4.2 Blast-SSAHA Hybridization

In this method of detecting fraud, the hybridization of BLAST and SSAHA algorithm is employed. It is therefore known as BLAH-FDS Algorithm. BLAST and SSAHA algorithm are very efficient sequence alignment algorithms therefore these two algorithms are used since the alignment of sequences is an efficient technique to examine the spending behaviour of customers. BLAH-FDS is a two-stage sequence alignment algorithm in which a profile analyser (PA) compares and determines the similarity of an incoming sequence of transactions on a given credit card with the genuine cardholder's past spending sequences. If there are any unusual transactions found by the profile analyser, then they are passed to a deviation analyser (DA) for any possible alignment with past fraudulent behaviour. The final judgment about the nature of the transaction is taken on the basis of the observations made by these two analysers.

When a transaction is carried out, the incoming sequence is merged into two sequences known as time-amount sequence (TA). The TA is aligned with the sequences that are related to the credit card in Customer Profile Database (CPD). This alignment process is done using BLAST SSAHA algorithm which increases the speed of the alignment process. If TA contains genuine transaction, then it would align well with the sequences in CPD. If there is any fraudulent transaction in TP, then mismatches occur in the alignment process. This mismatch produces a deviated sequence D which is aligned with Fraud History Database (FHD). A large similarity between deviated sequence D and FHD confer the presence of fraudulent transactions. PA evaluates a Profile score (PS) according to the similarity between TA and CPD. DA evaluates a deviation score (DS) according to the similarity between D and FHD. The FDM finally raises

an alarm if the total score (PS - DS) is below the alarm threshold (AT). The performance of BLAHFDS is good and it results in high accuracy. Also, the processing speed is fast enough for on-line detection of credit card fraud. It enumerates frauds in telecommunication and banking fraud detection. But it does not detect cloning of credit cards.

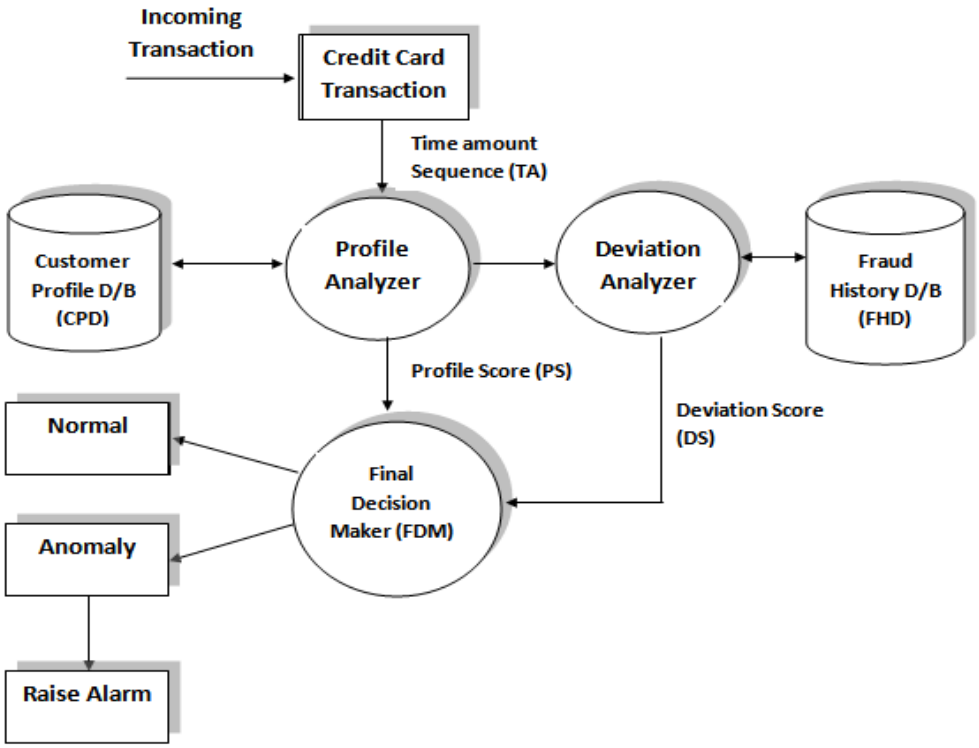


Figure 2: Architecture of BLAST-SSAHA Fraud Detection System

2.4.3 Hidden Markov Model

A Hidden Markov Model is a finite set of states; every state is associated with a probability distribution. Transitions among these states are administered by a set of probabilities called transition probability. In a specific state a possible outcome or observation can be produced which is associated symbol of observation of probability distribution. HMM categorizes card holder's profile as low, medium and high spending based on their spending behaviour in terms of amount. A set of probabilities for amount of transaction is being assigned to each cardholder. Amount of each incoming transaction is then matched with card owner's category, if it justifies a predefined threshold value then the transaction is decided to be legitimate else declared as fraudulent. HMM never check the original user as it keeps a log. The log that is maintained will also be a proof for the bank for the transactions that are made. HMM reduces the tedious work of an employee in bank since it maintains a log. HMM produces high false alarm as well as high false positive. HMM also works on human behaviour while doing online shopping.

2.4.4 Credit Card Fraud Detection using Genetic Algorithm

The Genetic algorithms are evolutionary algorithms which aim to obtain the better solutions to technically eliminate the fraud, a high importance have been given to develop secure and efficient e-payment system to detect whether a transaction is fraudulent or not. During a credit card transaction, the fraud has to be deducted in real time and the number of false alerts is being minimized by using genetic algorithm. The fraud that is detected is based on the customer's behaviour.

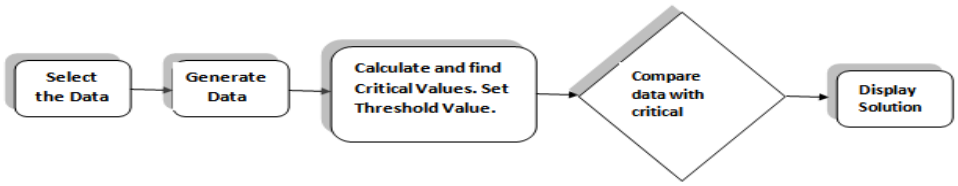


Figure 3: A Simple Method of Genetic Algorithm

Genetic algorithm procedure is repeated until a pre-specified number of iterations has passed, and the best solution is found. It is a parametric procedure and it should be problem undertaken to get a better performance. The list of the parameters and the settings are needed to generate fraud transactions. Such parameters are needed to compute the critical values, to calculate the Credit Card usage frequency count, Credit Card usage location, Credit Card overdraft, current bank balance, average daily spending etc.

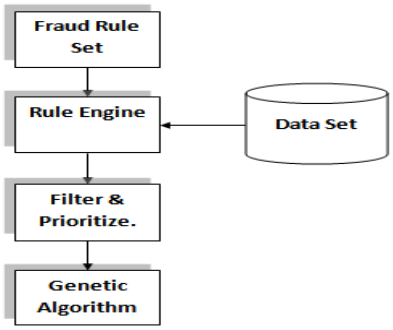


Figure 4: System Design

The aim is to obtain the better and optimal solutions. If this algorithm is applied to bank credit card fraud detection system, the probability of fraud transactions can be predicted soon after credit card transactions are done by the banks. And a series of anti-fraud strategies can be adopted to prevent banks from great losses before and reduce risks.

2.4.5 Stream Outlier Detection based on Reverse K-Nearest Neighbours (SODRNN)

SODRNN, standing for Stream Outlier Detection based on Reverse k Nearest Neighbours. This algorithm consists of two procedures: The Stream Manager and the Query Manager. Also, the whole window should be allocated in the memory. The Stream Manager receives the incoming data stream objects and efficiently updates the current window. When a new stream object comes, it only updates the knnlist and rknnlist of the influenced objects in the current window, in order to maintain current window perfectly rather than all the data stream objects in the current window. When the new incoming object is inserted, it needs only one pass of scan to the current window to find all objects whose k nearest neighbors are influenced. The updating of the knnlists of the influenced objects in the current window can update their rknnlists at the same time. The deletion of the expired object needs only update the rknnlists of the influenced objects in the current window according to its knnlist, and then update the knnlists of the influenced objects in the current window according to its rknnlist. When a user demands a query of the top n outliers, the Query Manager will make a scan of the current window and return n objects whose RNNk(p) is small as outliers of the query.

This algorithm reduces the number of scans to only one. Credit card validation checks and detects errors in a sequence of numbers therefore it detects valid and invalid numbers very easily.

2.4.6 Artificial Neural Network

Artificial Neural Networks (ANN) is applied for detecting fraud, mainly in the context of supervised classification. Artificial neural network (ANN) can be used in the recognition of characteristics timely and make predictions. The use of neural networking is motivated by the fact that it simulates the brain especially pattern recognition and associative memory. The neural network recognizes similar patterns, predicts future values or events based upon the associative memory of the patterns it has learned. These models are able to learn from the past and thus, improve their results as the time passes. They can also extract rules and predict future activity based on the current situation. By employing neural networks effectively, banks can detect fraudulent use of a card, faster and more efficiently.

In more practical terms neural networks are non-linear statistical data modelling tools. They are used to model complex relationships between inputs and outputs and to find patterns in data.

There are two phases in neural network:

a) Training phase and b) Recognition Phase.

Learning in a neural network is called training. There are two main types of Neural Network training methods:

a) Supervised and b) Unsupervised.

In supervised training, samples of both fraudulent and non-fraudulent records are used to create models. On the other hand, unsupervised training simply looks for those transactions, which are most dissimilar from the normal ones. Also, the unsupervised techniques do not need the

previous knowledge of fraudulent and non-fraudulent transactions in database. NNs can produce best result for only large dataset of transactions. And they need a long training dataset.

Two types of neural network are used in credit card fraud detection system:

- Back propagation neural network (BPNN): It is the most popular learning algorithm to train the neural network. It is a multi-stage dynamic system optimization method that minimizes the objective function. It is a supervised learning method and is a generalization of the delta rule. It is most useful for the feed-forward network which is network that has no feedback. It consists of three layers input, hidden and output layers. The incoming series of transactions passes from input layer through hidden layer and then to the output layer. This is also known as forward propagation. The input data is repeatedly feed to the neural network. With each presentation the output of the neural network is compared to the desired output and an error is computed. This error is then feed-back (back propagated) to the neural network and used to adjust the weights such that the error decreases with each iteration and the neural network gets closer to producing the desired output. This process is known as training. To train the NN so that it can be used for a credit card system last one- or two-year data is required of all the consumers. During training, the network is trained to associate outputs with the input patterns. After training when the network is used, it identifies the input pattern and tries to give output the associated output pattern. The power of neural networks is tested when a pattern that has no output associated with it, is given as an input. When credit card is being used by an unauthorized user the neural network-based fraud detection system checks for the pattern used by the fraudster and matches with the pattern of the original card holder on which the neural network has been trained, if it recognizes a pattern match, then neural network declares the transaction ok. However, this algorithm requires long training times, extensive testing, retraining of parameters, such as the number of hidden neurons, learning rate and momentum, to determine the best performance.

- Self-Organizing Map neural network (SOMNN): It is an unsupervised neural network learning method. In credit card fraud detection SOM has been used for forming customer profiles and analysing fraud patterns. In this method the transaction data is first identified and pre-processed. These data are fed in to SOM as input and weights of the neurons are adjusted iteratively. At the end of the training, the data is classified into genuine and fraudulent sets through the process of self-organization.

  This network contains two layers of node: an input layer and a mapping layer, in the shape of a two-dimensional grid. The purpose of these layers is to:

  a) classify and cluster the input data
  b) detect and derive hidden patterns in input data
  c) act as a filtering mechanism for further layers.

In this technique all transactions in the payment system are classified into genuine and fraudulent sets by following the two hypotheses:

1. If a new incoming transaction is similar to all previous transactions from the fraudulent set, then it is considered fraudulent.

2. If a new incoming transaction is similar to all previous transactions from genuine set, and then it is considered genuine.

2.4.7 DECISION TREES and SUPPORT VECTOR MACHINES:

Classification models which are based on decision trees and support vector machines (SVM) are developed and applied on credit card fraud detection problem. In this technique, each account is tracked separately by using suitable descriptors, and the transactions are attempted to be identified and indicated as normal or legitimate. The identification is based on the suspicion score produced by the developed classifier model. When a new transaction is proceeding, the classifier can predict whether the transaction is normal or fraud.

In this approach, firstly, all the collected data is pre-processed before we start the modelling phase. Since, the distribution of data with respect to the classes is highly imbalanced, so stratified sampling is used to under sample the normal records so that the models have chance to learn the characteristics of both the normal and the fraudulent record's profile. To do this, the variables that are most successful in differentiating the legitimate and the fraudulent transactions are founded. Then, these variables are used to form stratified samples of the legitimate records. Later on, these stratified samples of the legitimate records are combined with the fraudulent ones to form three samples with different fraudulent to normal record ratios. The first sample set has a ratio of one fraudulent record to one normal record; the second one has a ratio of one fraudulent record to four normal ones; and the last one has the ratio of one fraudulent to nine normal ones.

The variables which are used make the difference in the fraud detection systems. Our main motive in defining the variables that are used to form the data-mart is to differentiate the profile of the fraudulent card user from the profile of legitimate card user. The results show that the classifiers of SVM and other decision tree approaches outperform SVM in solving the problem under investigation. However, as the size of the training data sets become larger, the accuracy performance of SVM based models becomes equivalent to decision tree-based models, but the number of frauds caught by SVM models are still less than the number of frauds caught by decision tree methods.

2.4.8 FUZZY LOGIC BASED SYSTEMS:

2.4.8.1 Fuzzy Neural Network

The purpose of Fuzzy neural networks is to process the large volume of information which is not certain and is extensively applied in our lives. Syeda et al in 2002 proposed fuzzy neural networks which run on parallel machines to speed up the rule production for credit card fraud detection which was customer-specific. His work can be associated to Data mining and Knowledge Discovery in data bases (KD). In this technique, he used GNN (Granular Neural Network) method that uses fuzzy neural network which is based on knowledge discovery (FNNKD), to train the network fast and how fast a number of customers can be processed for fraud detection in parallel. A transaction table is there which includes various fields like the transaction amounts, statement date, posting date, time between transactions, transaction code, day, transaction description, and etc. But for implementation of this credit card fraud detection method, only the significant fields from the database are extracted into a simple text file by applying suitable SQL queries. In this detection method the transaction amounts for any customer is the key input data. This pre-processing of data had helped in decreasing the data

size and processing, which speeds up the training and makes the patterns briefer. In the process of fuzzy neural network, data is classified into three categories-

1. First for training,
2. Second for prediction, and
3. Third one is for fraud detection.

The detection system routine for any customer is as follows:

The detection system routine for any customer is as follows:
- Pre-process the data from a SQL server database.
- Extract the pre-processed data into a text file.
- Normalize the data and distribute it into 3 categories (training, prediction, detection)

For normalization of data by a factor, the GNN has accepted inputs in the range of 0 to 1, but the transaction amount was any number greater than or equal to zero because for a particular customer only the maximum transaction amount is considered in the entire work. In this detection method, there are two important parameters that are used during the training that are:

- Training error, and
- Training cycles.

With increase in the training cycles, the training error will be decreased. The accuracy of the results depends on these parameters. In prediction stage, the maximum absolute prediction error is calculated. In fraud detection stage also, the absolute detection error is calculated and then if the absolute detection error is greater than zero then it is checked to see if this absolute detection error is greater than the maximum absolute prediction error or not. If it is found to be true, then it indicates that the transaction is fraudulent otherwise transaction is reported to be safe. Both training cycles and data partitioning are extremely important for better results. The more there is data for training the neural network the better prediction it gives. The lower training error makes prediction and the detection more accurate. Higher the fraud detection error is, greater there is possibility of the transaction to be fraudulent.

2.4.8.2 Fuzzy Darwinian System

This technique uses genetic programming to develop fuzzy logic rules which are capable of classifying credit card transactions into “suspicious” and non-suspicious ones. It elaborates the use of an evolutionary- fuzzy system that is capable of classifying suspicious and non-suspicious credit card transactions. The developed system comprises of two main elements:

1. A Genetic Programming (GP) search algorithm and
2. A fuzzy expert system.

When the data is provided to the FDS system, the system first clusters the data into three groups namely low, medium and high which is known as fuzzy clustering. The genotypes and phenotypes of the GP System have some rules which match the incoming sequence with the past sequence. Genetic Programming is used to develop a series of variable-length fuzzy rules

that characterize the differences between classes of data placed in a database. The system is developed with the definite aim for insurance-fraud detection which includes the challenging task of classifying the data into the categories: safe and suspicious. For classification of transactions, when the customer's payment is not overdue, or the overdue payment is less than three months, the transaction is considered as "non-suspicious, otherwise it is considered as suspicious.

The Fuzzy Darwinian detects suspicious and non -suspicious data easily and also detects stolen credit card Frauds. This system has very high accuracy and produces a low false alarm in comparison with other techniques, but it is highly expensive. The speed of the system is also low.

2.4.9 Fraud Detection using Meta-Learning:

Meta-learning is a strategy that provides the means of learning of how to combine and integrate a number of separately learned classifiers or models into one. Therefore, a meta-classifier is trained relatively with the predictions of the base classifiers. This system has two key component technologies:

1. Local fraud detection agents that learn how to detect fraud and provide intrusion detection services within a single collective information system, and
2. Meta-learning system that combines the collective knowledge attained by individual local agents. This is a secure and integrated system.

Once derived local classifier agents or base classifiers are produced at some sites, two or more such agents may be composed into a new classifier agent i.e. a meta-classifier by a meta-learning agent. This meta-learning system will allow financial institutions to share their models of fraudulent transactions by exchanging classifier agents in a secured agent system without disclosing their patent data. In this way their competitive and legal restrictions can be met, and they can still share information.

3 Model Analysis

3.1 Comparative Study

There are various methods of detecting a credit card fraud. We've presented a comparative study of some of the fraud detection methods based on credit card. If one of these or combination of algorithms is put into practical use for bank credit card fraud detection system, the probability of fraud transactions can be known in advance soon after the credit card transactions are carried by the banks. A series of anti-fraud strategies can be adopted to prevent the banks from great losses sooner and reduce the risks. This paper gives contribution towards the effective ways of credit card fraud detection. A comparison table is prepared to compare various credit card fraud detection mechanisms based on some parameters such as, accuracy, speed and cost.

Table 3: Performance Measures of various data mining algorithms

Models	**HMM**	**FDS**	**DST & BN**	**BSH**	**GA**	**SODRN**	**ANN**	**Meta-learning**	**FNN**	**SVM**	**SOM**
Accuracy	Low	Very High	High	High	Medium	Medium	Medium	High	Good	Medium	Medium
Speed of Detection	Fast	Very low	Low	Good	Good	Good	Fast	Low	Very fast	Low	Fast
Cost	Very Expensive	Very Expensive	Expensive	Moderate	Inexpensive	Expensive	Expensive	Expensive	Expensive	Expensive	Expensive

From the above table we realise that BSH is optimally the best method to detect credit card fraud as its Accuracy is High, Speed of Detection is Good, and Cost is Moderate. This will allow the bank to adopt it with no hassles with respect to the above three factors. For future work we can work on the implementation time required by each of the above models which will also give us a clear understanding if the model is suited for a particular bank or not.

All these techniques of credit card fraud detection discussed, have its own weaknesses as well as strengths. Thus, this survey enables us to create a hybrid approach for developing some effective algorithms which can perform well with minimum costs and higher accuracy.

ABBREVIATIONS:

HMM- Hidden Markov Model

FDS- Fuzzy Darwinian System

DST- Dempster-Shafer Theory

BN- Bayesian Network

BSH- Blast-Ssaha Hybridization

GA- Genetic Algorithm

SODRNN- Stream Outlier Detection Based On Reverse K-Nearest Neighbors

ANN- Artificial Neural Networks

FNN- Fuzzy Neural Network

SVM- Support Vector Machine

SOM- Self Organizing Map Neural Network

Figure 5: Abbreviations used in the table above

4 Case Study:

The company of this case study is one of the leading online luxury fashion retailers in 2016. It follows a marketplace business model, selling items from more than 400 partner boutiques on a commission basis. The value proposition for boutiques includes payment pro-cussing, branding, online content creation, out-bound logistics and customer service. As part of payment processing, fraud detection is included as an added value service provided. Moreover, being acknowledged by customers and card associations as a trustworthy merchant is crucial when doing business online. Fraud detection involves a fundamental trade-off. On the one hand, the company has to minimize the level of fraud, maximizing the detection of fraudulent transactions, thus avoiding chargebacks. On the other hand, it must provide high payment acceptance rates in order to convert as many sales as possible and minimize the number of customer insults (i.e. the number of legitimate transactions refused). At the moment, the company relies on expert-rules to perform a first check of incoming orders, but the majority of the orders (around 65%) are manually verified by the order approval team for any indications of fraud. This is time consuming and is not sustainable, as the company's prospects for the mid-term future foresee a continued growth rate around 50–70% per year. Increasing the automation level is thus essential to quickly evaluate a large and growing number of orders. This high growth rate introduces some nuances to the problem, since the history of orders to study from is skewed towards more recent dates. Moreover, the company ships to everywhere in the world, which increases the difficulty of preventing fraud.

The company tracks several key performance indicators (KPIs)related to fraud detection:

- The automation level – percentage of orders automatically processed;
- Chargeback level – percentage of orders which originate chargeback;
- Rate of refused payments – percentage of payments which are refused; and
- Speed of processing – time it takes to approve or reject an order payment.

The objective of the company is to develop a system which will result in a higher automation level, while not increasing the charge-back level and the rate of refused payments. In the medium-term the new process should allow the company to reach a level of automation of 80%, with a chargeback level under 1% and a refused payment rate under 4.5%.

The proposed approach consists in building a risk scoring system based on machine learning methods which will estimate a Fraud Suspicion Score for each order. A diagram of the proposed process can be seen in Figure above. The suspicion score estimated by the model should be a number between 0 and 1. The risk scoring system would also evaluate whether the score falls below a certain threshold (e.g. lower than 30%), where the order would be automatically approved. Orders with a higher score would be manually revised by the order process-in team, which could also count on the suspicion score for a better evaluation.

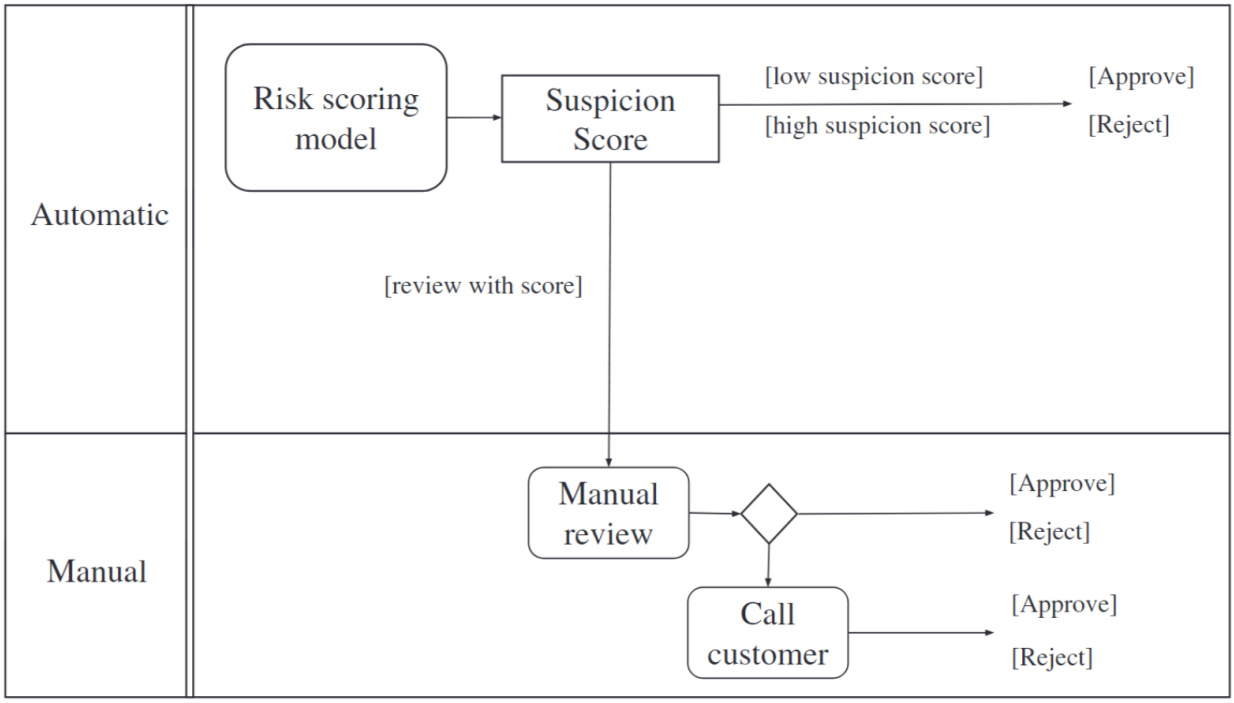


Figure 6: Proposed framework for credit card processing

4.1 Data mining techniques for fraud detection

After doing an in-depth research in the field of data mining for fraud detection above in this study, we chose to use methods of supervised learning for the classification problem, because it is common for fraud detection applications to have labelled data for training. Three different models were chosen to test this approach. Logistic regression because of its popularity, and random forests and support vector machines, which have been used in a variety of applications showing superior performance. Support vector machines have been shown to perform well in classification problems and random forests are very attractive for fraud detection due to the ease of application and being computationally efficient.

4.2 Data understanding and preparation

Building the dataset on which to base the study is not a trivial activity and involves decisions which can greatly affect the quality of the data mining project. Moreover, one must transform all categorical variables into numerical, in order to use machine learning algorithms such as support vector machines. This section describes the reasoning behind the building of the data set for this project. Handling the data was done with Python version 3.6. Two particular modules for Python, Pandas data structures and Scikit-learn, were especially useful for running the algorithm.

4.3 Data collection

Taking into consideration the input from the fraud analysts at the case study company and the examples in the literature, we decided that the unit of analysis would be each individual order. With that in mind, the objective at this stage was to build a data set where each row corresponded to one order and the columns represented different attributes of such order. Since a substantial part of the orders has multiple products, their attributes had to be aggregated (e.g. number of items of each gender and product family). Additionally, we merged data about the payment processing (e.g. AS and CVV responses) and the customer's behavior online before the purchase (e.g. number of page views).
Another key aspect in classification is the definition of the class for each data entry. In our case, we faced a binary classification problem where each data entry – an order – has to be labelled as belonging to one of two classes: fraudulent or non-fraudulent.

Two criteria are taken into consideration when defining the classes of the dataset and we created a variable which is called Label Fraud with a value of 0 or 1, such that:

1. An order that originated a chargeback or was manually marked as fraudulent was labelled as fraudulent (Label Fraud= 1); and
2. All other orders were labelled as legitimate transactions (Label Fraud=0).

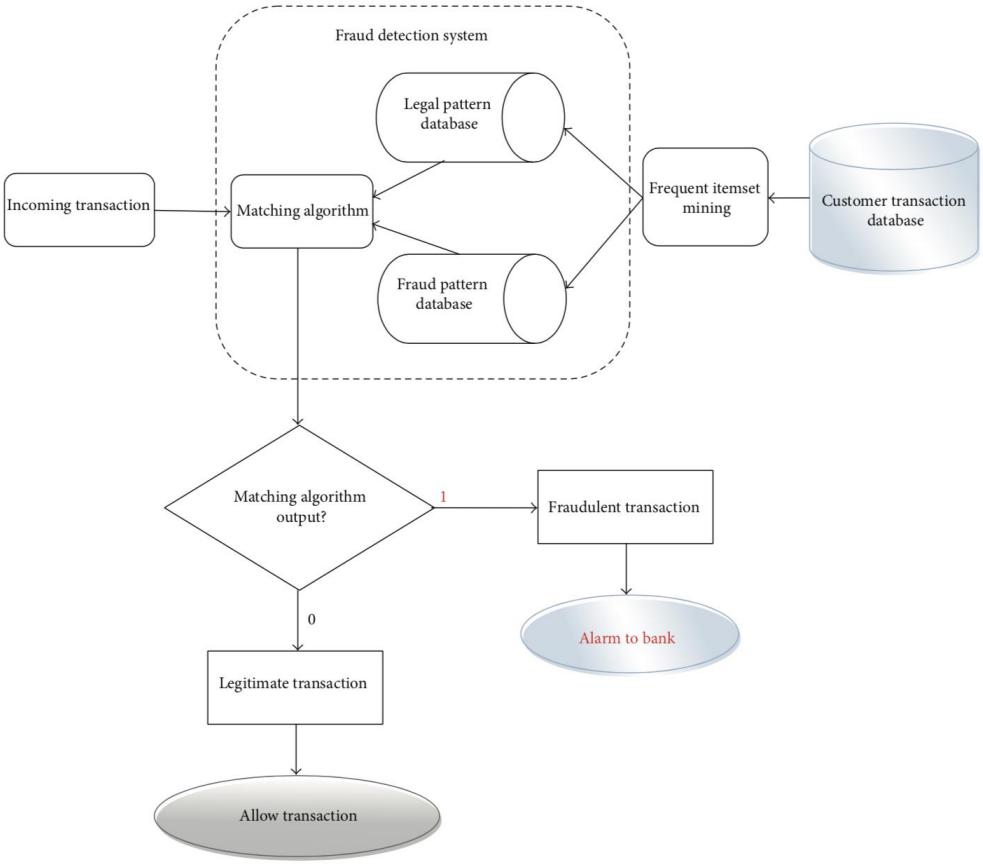


Figure 7: Proposed Automatic framework for credit card processing

Not all data can be labelled, due to the long time it takes for the merchant to realize that a fraudulent transaction has occurred. Most commonly, the alert is first raised by the legitimate owner of the credit-card, who notices a transaction in his bank statement that he did not make. Then, he must place a complaint on that transaction, which will originate a chargeback. It seems natural that such lengthy procedures take several weeks between the time that the fraudsters use the credit-card and the merchant is notified of the chargeback. This restricts the orders we can use for data mining, because the most recent frauds will not have been detected yet. In the past, 80%of the chargebacks at this merchant arrived within the first 11 weeks after the transaction. Knowing this, it was decided that the data to reanalyzed should be no more recent than 16 weeks.

The collected data is split in two: a training and a testing set. The training set is then used for the algorithm to learn how to classify orders. The testing set is later used to validate the effectiveness of the algorithm, removing the overfitting effect, i.e. the increase in performance that the algorithm has on the instances it has based its learning, as it describes not only the underlying relationship, but also random error (present in those particular instances).

We chose to consider the most recent 20% of orders for testing. Dividing by date was due to the fact that no pattern of fraud could be learned prior from the date, but fraud behavior may change. Therefore, this approach will give a conservative performance estimation, as more recent orders should better emulate new ones. The result of the division is a training set containing the orders placed between 2015-01-01 and 2015-06-21 (totaling 347,572 observations), and a testing set containing the orders placed between2015-06-22 and 2015-08-04 (in a total of 86,893 orders).

4.4 Variables' transformations

Categorical variables with few categories were transformed into several features through one-hot-encoding, which consists of encoding a variable through binary numbers. In our case, we created one column for each of the categories of the variable. This technique will allow the model to fit such variables, but as the dimensionality grows the model loses generality. Hence, one-hot-encoding should not be applied to variables with an extensive number of categories. Variables including too many categories, such as the address country (more than 200 different countries), should not be transformed into individual features through one-hot-encoding.
Therefore, we chose to cluster countries by fraud risk, calculated by the ratio of fraudulent orders over the total number of orders in that country "i":

$$Fraud\ ratio\ i = \frac{Fraudulent\ Orders}{Total\ Orders}$$

The countries have been clustered in four groups. Countries with less than 30 orders, have been considered not significant and attributed an intermediate level of risk. When transforming an entry for a new order, we check which risk-level had been assigned to that country in the training phase. Had such a category not been assigned a risk-level before (e.g. a country where no order from the training set was shipped to), the risk-level is set to the intermediate level.

In order to get more significant variables to train the models, we engineered new variables through abstraction and combination of variables. The first engineered variables were created to riper-sent the degree of similarity between certain pairs of categorical variables. We created a binary function which generated a new feature with the value of 1 in case of a match and 0

in case of no match. This function was applied to the pairs of variables {(billing country, shipping country), (billing country, card country), (shipping country, card country)}. These were all predefined country fields, which ensured the match could be done by a simple comparison of strings.

On the other hand, the variables referring to names (e.g. name on card, user name) would often not match because the names were spelled in different order. Hence, we used another function which calculates the similarity between two names. This function is based on n-gram similarity and its output is a continuous value in [0,1]. It was applied to the pairs of variables {(Username, Card Name), (billing city, shipping city), (billing zip-code, shipping zip-code)}.

A function which confirms that the customer has entered a valid telephone number was also created, by checking whether the input of the user was a number. Lastly, a function summarizing the customer's recent buying behavior was created. This function counts the number of orders placed by this customer in the N days prior to the current purchase (with N= 10 being chosen for the best performance). The final list of 70 features (categorical variables are not exhaustively listed), plus the label, can be seen in table below.

Table 4: Important attributes and their possible values used for prediction

Name	**Values**
Label Fraud	0, 1
OrderTimeGMT	0, 23
OrderValue	R>0
C2VCode_Match	0, 0.5, 1
AVSCode_Match	0, 0.5, 1
Quantity	Z>0
CategoryClothing	Z>=0
CategoryOther	Z>=0
GenderMen	Z>=0
GenderWomen	Z>=0
GenderOther	Z>=0
Brand_1	Z>=0
BrandOther	Z>=0
Currency:AUD	0, 1
Currency:USD	0, 1
PaymentType_1	0, 1
PaymentType_2	0, 1
PaymentType_3	0, 1
PaymentType_4	0, 1
NumcardsUsed	Z>=0
PaymentAttempts	Z>=0
ValidUserPhone	0, 1
BillCountry = ShipCountry	0, 1
BillCountry = CardCountry	0, 1
ShipCountry = CardCountry	0, 1
BilllRegion=ShipRegion	0, 1
TimeSince FirstOrder	R>=0

Similarity(UserName/CardName)	0, 1
Similarity(BillCity/ShipCity)	0, 1
Similarity(BillZip/ShipZip	0, 1
ShipCountry_RiskLevel	1 ,.., 4
ShipCity_RiskLevel	1 ,.., 4
ExpressShipping	0, 1
NumOrderLast10DaysSameUser	Z>=0

In order to get the best performance of the machine learning algorithms, the data must be clean and complete. We chose to complete missing data entries by imputing a value in the missing variables. In the case of categorical variables, a missing value would correspond to a zero in each of the corresponding columns after the one-hot-encoding transformation. In the case of numerical variables, the missing value was replaced by the mean of the sample of that variable in the training set.

Variables which are measured in different units such as Quantity or Order Value were standardized to values between 0 and 1, so that they can more easily be compared. Among the several techniques for standardization, we chose to use Min-Max. This technique was preferred over the commonly used Z-value standardization, because it transforms variables to the range [0,1], which is consistent with the range of the many binary variables in the model.

4.5 Modelling and evaluation

This section describes the selection of the model and parameters to use for the order classification.

4.5.1 Performance Measures

Before evaluating the results of each model, we had to decide on the performance measures to compare. For each observation X, we have an associated real class label from the set {0,1} and a corresponding predicted label. Observations which are classified correctly as belonging to class 1 are called true positives, while observations correctly classified as class 0 are called true negatives. There are two other possible outcomes in which the prediction incurs in an error. The type 'I' error occurs when the positive class is predicted, but the observation label is 0 (these predictions are called false positives). The type 'II' error occurs when the prediction of class 0 does not agree with the observation's true class which would be 1 (false negatives). The two types of error can have different implications. In the case of fraud detection, a false positive error would correspond to a legitimate observation being labelled as fraud. A false negative error would happen in the case of a fraudulent transaction being classified as legitimate.

In the case of classifiers such as random forests, the output is a continuous value that correlates with the probability of the observation belonging to class 0 or 1. Therefore, a threshold must be defined in order to determine the final class (0 or 1), based on the real value obtained with the classifier. The receiver operating characteristic curve (ROC) offers a way of visualizing different outcomes describes ROC curves as depicting the relative trade-offs between benefits (true positives) and costs (false positives), working very well in practice as a general measure

of classifier performance. Observations with a score under the threshold are classified as class 0, while a score above the threshold would predict that the observation belongs to class 1. A ROC curve plots the true positive rationales positive rate for each threshold between$-\infty$and $+\infty$. The area under the curve (AUC) is equivalent to the probability that the classifier will give a higher score to a randomly chosen observation of class 1 than to a randomly chosen observation of class 0.

Precision-recall (PR) curves have been used as an alternative performance measure to ROC. In a PR graph, we plot the precision and recall for each threshold. One of the conclusions of this study is that optimizing the AUC-ROC will not guarantee the best result in AUC-PR. Therefore, we chose to use AUC-PR as the last measure of comparison between the performance of the different hypothesis in cross-validation.

4.5.2 Cross Validation

From the three chosen families of models (logistic regression, support vector machines and random forests), we want to know which can make the best predictions. Moreover, we want to deter-mine the best hyperparameters for each model.

The whole data set was previously split into training and testing sets. Now a further split is performed on the training set. Part issued in actual training, while the rest, which we call the validation set, is used in finding the best parameters of each model. Wean estimate the performance by repeating the training-validation procedure multiple times. The most commonly applied validation technique is called-fold cross-validation, in which the training set is divided in K folds, with each of the folds being left out at a time for the training-validation sequence

4.5.3 Results of Cross Validation

We applied 10-fold cross-validation for the three models: ran-doom forests, support vector machines and logistic regression. The choice of the parameters to test was done by grid search, a tech-unique which consists of performing an exhaustive search through a predefined space of parameters. In some cases, a refined grid search was additionally performed after finding a suitable subset of the parameter space.

We have also compared results when training on a balanced sample (i.e. equal number of fraudulent and legitimate records) achieved by randomly under-sampling the legitimate records vs. an unbalanced sample of all records. Each training set had 347,572 records of which 312,814 were used for training and 34,758 for validation at a time. In the case of under-sampling, the records used for train-in were reduced to a number around 13,000 (circa two times the number of fraudulent cases in nine tenths of the records), while the validation was done on all 34,758 records from the validation fold. Random forests and logistic regression were validated with both balanced and unbalanced data sets. Support vector machines has a much higher computational complexity and therefore it was just trained on the balanced sets generated by under-sampling.

4.5.4 Random Forests

Random forests are an ensemble method which consists of creating many decision trees and combining their estimates. The parameters which were varied were the minimum number of features to split each node of the tree (Min. Split), the criterion for quality of node split (Gini impurity or entropy), the number of trees, and whether a balanced set was used (under-sampling). The number of trees is not a true parameter, as more trees will always lead to a higher performance, but also more computational time. Nevertheless, with 1500 trees, the performance was practically stagnating, so this was the maximum value we tested. The top five results of this grid search can be seen in the table below. Under-sampling did not lead to a better performance. In fact, there was no indication that using a balanced sample by under-sampling would achieve a different performance. Regarding the criterion for split, itis clear that the best performance is achieved by using the entropy criterion.

Table 5: Grid search Results for Random Forrest

Num Trees	Min. Split	Criterion	under-Sampling	AUC-PR	AUC-ROC
1500	10	Entropy	No	0.479	0.935
1500	12	Entropy	No	0.477	0.935
1500	11	Entropy	No	0.475	0.935
1500	6	Entropy	No	0.475	0.935
1500	16	Entropy	No	0.474	0.935

4.5.5 Support Vector Machines

Support vector machines is an advanced statistical classifier, which can make use of a kernel trick to map the data to a high dimensional feature space. We used an implementation of support vector machines which yields a continuous probabilistic output. We employed a radial basis function kernel (RBF). The kernel coefficient Gamma and regularization term were varied in a logarithmic scale. A high regularization term represents a weaker regularization. Thebes results were obtained for a value of= 10, as can be seen from Table 2.

Table 6: Grid Search Result for Support Vector Machines

C	Gamma	Kernel	Under-Sampling	AUC-PR	AUC-ROC
10	0.0464	RBF	Yes	0.337	0.906
10	0.0215	RBF	Yes	0.336	0.902
10	0.01	RBF	Yes	0.319	0.896
10	0.1	RBF	Yes	0.317	0.903
10	0.1	RBF	Yes	0.305	0.895

4.5.6 Logistic regression

Logistic regression is a widely used method for classification and regression. Due to memory limitations, we could not train a logistic regression on a higher order representation of the features (e.g. second or third order transformation). Like in support vector machines, we varied the regularization term. The only solver used was an implementation of the Limited-memory Brayden-Fletcher-Goldfarb-Shannon (lbfgs) optimization algorithm.

Table 7: Grid Search Results for logistic Regression

C	Solver	under-Sampling	AUC-PR	AUC-ROC
100,00	lbfgs	No	0.36	0.907
3.16	lbfgs	No	0.357	0.905
1	lbfgs	No	0.349	0.902
0.32	lbfgs	No	0.324	0.895
3.16	lbfgs	Yes	0.313	0.903

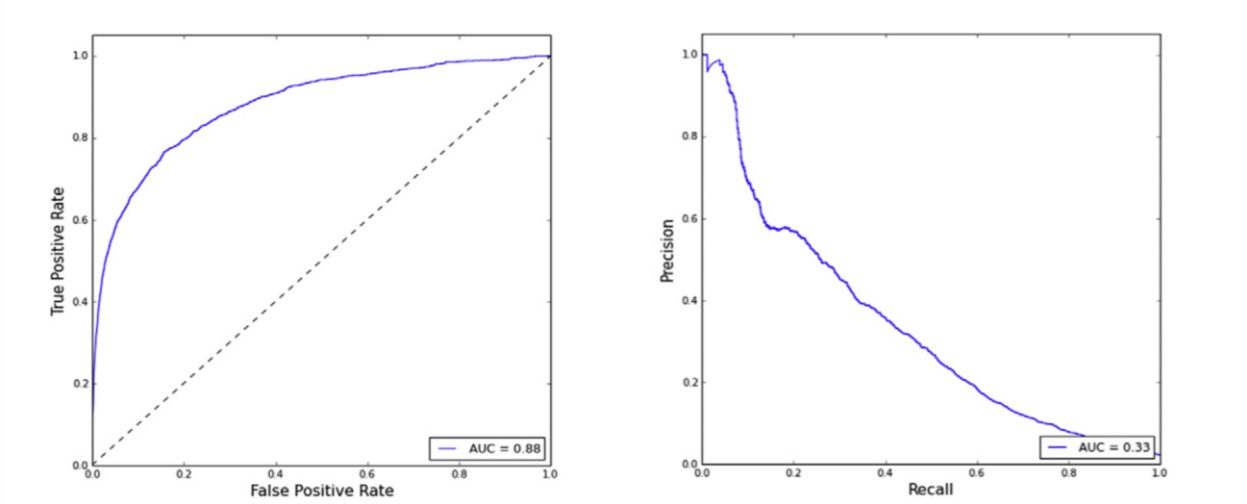
4.5.7 Validation Results

When looking at the cross-validation results in Tables above, we conclude that random forests achieved the highest performance. The performance of support vector machines and logistic regression is similar, with values of AUC-ROC slightly lower than random forests. However, the value of the primary performance measure, AUC-PR, of the best random forests model is considerably higher than for the other two models. Therefore, random forests were the selected model.

4.5.8 Testing results

When testing, we want to estimate the performance of the model, which we chose in training – random forests, in evaluating new data. The performance in cross-validation has an optimistic bias, thus we expected to get a lower performance in testing. Table shows the results of testing. Contrary to the performance results in validation. The value of the area under the precision-recall curve (AUC-PR) was not so satisfactory. However, random forests still presented a high value for the area under the receiver operating characteristic curve (AUC-ROC).

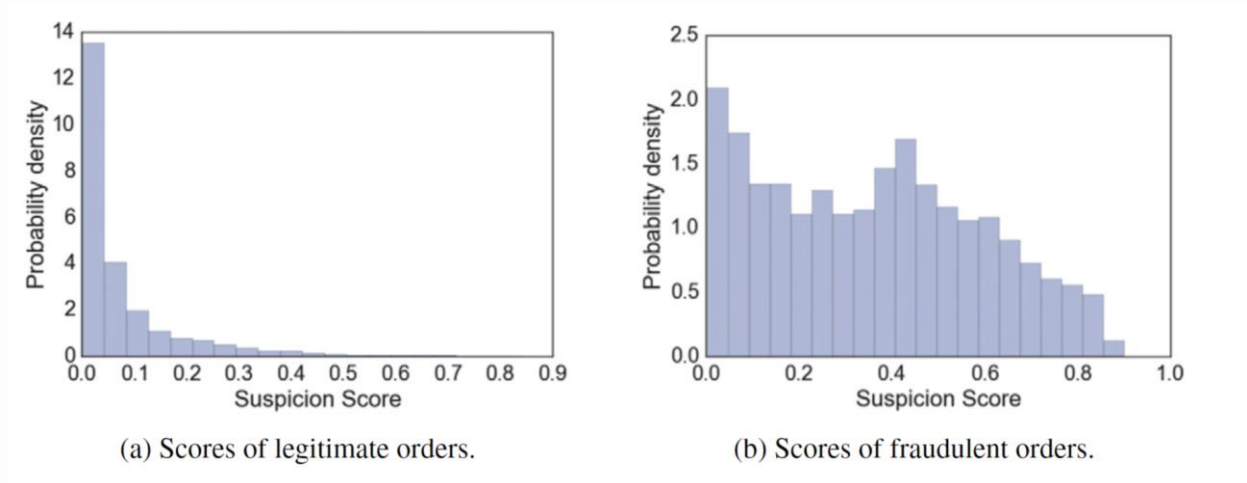
Classifier	AUC-PR	AUC-ROC
Random Forrest	0.333	0.880



(a) ROC curve. (b) Precision-Recall curve.

Figure 8: ROC and PR curves of Testing Results with random forest

The distribution of scores for legitimate and for fraudulent orders can be seen on the histograms in figure below. On the left, we see the distribution of the scores which were estimated for legitimate orders. On the right the scores estimated for fraudulent orders. The first thing to notice is that the distributions are different, which is a sign that the classifier recognized the two classes. There is a high occurrence of low scores for legitimate orders, which is what we would expect. On the other hand, fraudulent orders have a rather even distribution of scores. It would be expected that more fraudulent orders had high scores (near the value 1). This is likely to be a consequence of the unbalanced data set used for training, as the classifier can very well identify legitimate orders but has more difficulties with fraudulent observations.



(a) Scores of legitimate orders. (b) Scores of fraudulent orders.

Figure 9 Histograms of score prediction for each of the labels

In order to calculate business metrics such as the number of chargebacks (false negatives) and the number of refusals of legit-mate payments (false positives), we must define a score threshold for considering a transaction legitimate or fraudulent based on its suspicion score. We detail two different approaches to defining this threshold, the first approach poses a scenario where manual revision is not possible and all orders are automatically approved and rejected. The second approach explores the possibility of combining automatic classification with manual revision.

5 Decision Support Analysis

5.1 Defining a score Threshold – Approach 1

This approach estimates the performance in the case that all orders are automatically approved or rejected in accordance with their suspicion score. One approach to define the threshold is to look for the value which originates the point of the curve closest to the top left corner in the ROC space. This is the point where **Recall** ($\frac{True\ Positives}{\text{TruePositives+FalseNegatives}}$) and **Specificity** ($\frac{True\ Negatives}{\text{True Negatives+False Positives}}$) have the same value, which achieves the compromise between the two measures. Alternatively, different weights could be attributed to Recall or Specificity to find another threshold. Setting the threshold at this value, we can draw the confusion matrix of results which can be seen in Table Below:

Table 8: Confusion matrix When Setting Threshold at c = 0.22

	Legitimate (Actual)	**Fraud (Actual)**	**Total**
Legitimate (Predict)	77129 (88.76%)	618 (0.71%)	77, 747
Fraud (Predict)	7904 (9.09%)	1242 (1.42%)	9146

In case this classifier would be processing all orders, the estimated number of chargebacks would be 0.71% (number of false negatives), while 1.42% of all orders would be correctly rejected as fraudulent. However, we would incur in a high number of false positives (9.09%). This would lead to more than 10% of all orders being rejected. This is not acceptable in practice, and thus manual revision is still necessary for the orders considered fraudulent by the classifier. This approach is explored in the next section.

5.2 Defining a score threshold - Approach 2

Our second approach consists in bringing together automatic classification with manual revision of orders. Hence, all orders with score below a specific threshold are automatically approved, while the rest of the orders are manually revised. The specific threshold can be chosen by evaluating the trade-off between the cost of manually revising an order and the financial risk of fraud. At our case study company there were already resources available to revise most orders manually. In accordance with the company's objectives, the goal that only 20% of orders would be manually revised. Hence, the threshold was set at the value which is higher than 80% of the score of all orders. The advantage of this approach is that it can be easily adapted to new business requirements. If the company is interested in taking a lower risk, it can choose to decrease the number of automatically approved orders, incurring in higher manual revision costs. The Figure below depicts this trade-off by showing how many fraudulent orders will be automatically approved if we vary the level of automation.

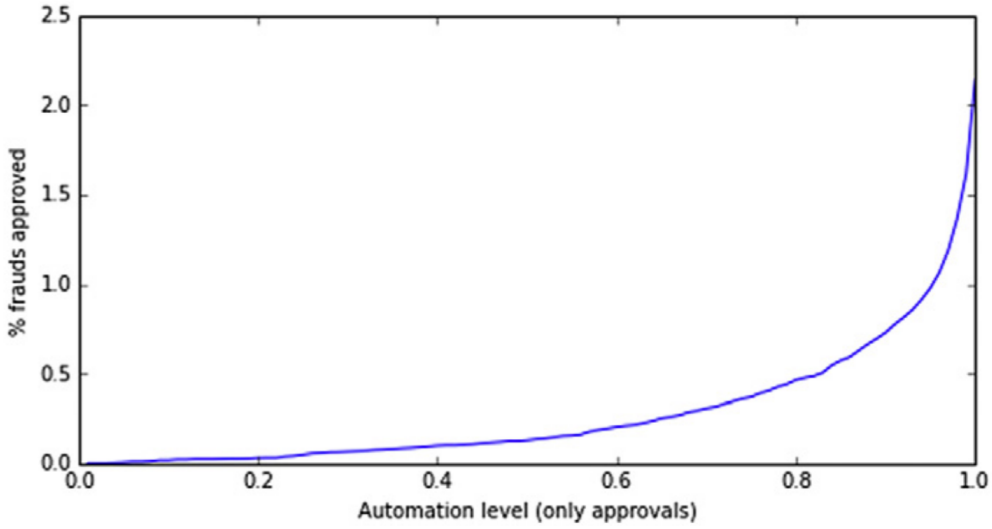


Figure 10: Plot of automation level and fraud

In order to build a confusion matrix, we must estimate the effectiveness of manual revision. The following assumptions were made, based on past performance of manual revision at this company:

- The score threshold splits the orders which would be auto-magically approved (score under the threshold) from the ones which would be revised (score above the threshold);
- When a fraudulent order is manually revised, there is a 75%probability that it will be refused; and
- When a legitimate order is manually revised, there is a 90%probability that it will be accepted.

The Table below shows the results of this approach. The results show a high value of Specificity, which can be interpreted as the fraction of legitimate orders which are approved, in this case circa 98%. On the other hand, the Recall value is rather low: only circa 59% of the fraudulent orders would be refused. Precision is the fraction of fraudulent orders out of all which were refused $\left(\frac{True\ Positives}{TruePositives+FalseNegatives}\right)$ Only 41% of the refused orders were actually fraudulent. Fallout can be interpreted as the "false alarm" rate, i.e. the probability of falsely rejecting a legitimate order$\left(\frac{True\ Negatives}{True\ Negatives+False\ Positives}\right)$. This value is equal to 1 minusthe value of specificity, in this case circa 2%. This combination of therandom forests classifier with the manual revision of orders, seems toyield very good results in increasing the automation level and having low rate of customer insults (fallout). These values meet all the goals of the company.

Table 9: Confusion matrix When Setting Threshold at c = 2

	Legitimate (Actual)	**Fraud (Actual)**	**Total**
Legitimate (Predict)	83441 (96.02%)	768 (0.88%)	83914
Fraud (predict)	1592 (1.83%)	1092 (1.25%)	2684

Table 10: Performance measures When Setting Threshold at c = 2

Automation level:	80%
Recall:	0.587
Specificity:	0.981
Fallout:	0.019
Precision:	0.407

5.3 Variable Importance

The random forests model allows us to estimate the relative importance of each of the features. We can see the importance of the top 10 attributes in Table below. Attributes which were represented by many features such as the product brand have been aggregated intone feature in order to have a better overview of their importance. The importance of each variable is calculated as the "mean decrease impurity".

Table 11: Relative Importance of the top 10 Attributes

Importance	Feature	Description
13.3%	Time Since First order	Time elapsed since user's first purchase at this merchant
10.3%	Order Value	Value of Order
7.6%	Brand	All the features describing the product brand
6.3%	ShipCity_RiskLevel	Engineered variable - risk level associated with shipping city
6.1%	Quantity	Quantity of items in the order
5.9%	Gender	Gender of items in the order
5.4%	Similarity (UserName/CardName)	Engineered variable - similarity between username and name on credit card
4.9%	Name Cards Used	Number of cards used by customer at this merchant
4.9%	Payment Attempts	Payment attempts by user for this order
4.6%	Order Time GMT	Absolute time at which the order is placed

We can see that the time since the customer's first order is the most important attribute. The value and quantity of items in the order also appear to be relevant features. On the product perspective, its brand and gender combined have the same level of importance as the customer's first order date. The risk associated with the city where the order will be shipped to is the fourth most important attribute. A few features related to the payment itself also show sig-indicant importance. These include the similarity between the name on the credit-card and the name of the customer, the number of cards used by the customer in all his orders at this merchant, and the number of repeated payments attempts for this order.
Other variables such as the currency used or the similarity between billing and shipping addresses ranked lower in terms of relative importance. It should be noted that in some cases, particular variables will completely fail to identify fraud. However, their com-bination proved to be effective. The information we get from this analysis can be of use to improve the manual revision process and inbuilding new models in the future.

6 Conclusion and Future Work

This report addressed all the data mining techniques used in the banking sector with a special in-depth literature review of Fraud detection in Credit Cards. The case study deals with the designing, developing and implementing a risk scoring system (using datamining techniques) at an e-tail merchant. This report can help researchers and practitioners to design and implement data mining-based systems, as it describes the complete development process and addresses practical implementation issues.

From this case study it is clear that the choice of which variables to use is very important. Our exploratory analysis will provide insights to the fraud analysts for improving their manual revision process. In that sense, it is important to explore the database and understand what other variables could be used that were not obvious at first. The most relevant features for fraud detection turned out to be those which were engineered out of one or multiple base variables. The time since the customer's first order, the similarity measures between names or the grouping of cities by risk are examples of such features. Future work comparing different transformation functions and providing guidance on which features to engineer would be very useful for this area.

Concerning the core part of the system, which is the classifier, we have realized that supervised learning methods are applicable to fraud detection in an e-tail merchant setting. All the three machine learning algorithms (logistic regression, support vector machines and random forests) have provided good results. Random forests achieved the highest performance of the three. This algorithm seems to be very adequate to be used for fraud detection, not only because of its good performance, but also due to the ease of implementation and fast computation time even with large datasets. We concluded that the use of balanced set of observations (under-sampling legitimate records) was not significantly different in performance than the use of much bigger unbalanced full set of records. The testing results with the random forests algorithm showed that such a model would perform well enough to be of practical use. The advantages of a data mining approach are the possibility of automatically processing a large volume of orders, allowing e-tail businesses to expand sustainably. When using a continuous classifier such as random forests, it is critical to choose the score threshold for considering an order to be legitimate or fraudulent. This approach assumes that the merchant will manually revise orders with a score above the threshold, instead of directly cancelling them. The merchant is provided with the possibility to define the threshold by choosing the share of orders which should be automatically processed. We suggest as future work to explore whether oversampling of fraudulent records would lead to a different conclusion.

From the case study we realized how we could implement the fraud detection system in a bank or any financial institution. We clearly understand the predictive algorithms we could use to get optimal results. This report is a good starting point for any fraud analyst to understand credit card fraud and the techniques in place to tackle it.

7 References

Chan, S.W. & Franklin, J. 2011, 'A text-based decision support system for financial
sequence prediction', Decision Support Systems, vol. 52, no. 1, pp. 189-98.

Chitra, K. & Subashini, B. 2013, 'Data mining techniques and its applications in banking sector', International Journal of Emerging Technology and Advanced Engineering, vol. 3, no. 8, pp. 219-26.

Jha, S. & Westland, J.C. 2013, 'A descriptive study of credit card fraud pattern',
Global Business Review, vol. 14, no. 3, pp. 373-84.

Mahmoudi, N. & Duman, E. 2015, 'Detecting credit card fraud by modified Fisher discriminant analysis', Expert Systems with Applications, vol. 42, no. 5, pp. 2510-6.

Nuno Carneiroa, G. F. b. M. C., 2017. A data mining based system for credit-card fraud detection in e-tai. *Decision Support Systems*, pp. 91-101.

Quah, J.T. & Sriganesh, M. 2008, 'Real-time credit card fraud detection using computational intelligence', Expert systems with applications, vol. 35, no. 4, pp. 1721-32.

Suraj Patil, V. N. P. S., 2018. Predictive Modelling For Credit Card Fraud Detection Using Data Analytics. In: *Procedia Computer Science 132 (2018).* s.l.:International Conference on Computational Intelligence and Data Science (ICCIDS 2018), pp. 385 -395.

SurveyIvan Herman, M. C. S. G. M. o. a. M. S. M., 2000. Graph Visualization and Navigation inInformation Visualization: A Survey. *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS*, p. 24.

Seeja, K. & Zareapoor, M. 2014, 'FraudMiner: A novel credit card fraud detection model based on frequent itemset mining', The Scientific World Journal, vol. 2014.

Sharma, A. & Panigrahi, P.K. 2013, 'A review of financial accounting fraud detection
based on data mining techniques', arXiv preprint arXiv:1309.3944.

Sethi, N. & Gera, A. 2014, 'A revived survey of various credit card fraud detection techniques', International Journal of Computer Science and Mobile Computing, vol. 3, no. 4, pp. 780-91.

Thiago Poleto(✉), V. D. H. d. C. a. A. P. C. S. C., 2015. The Roles of Big Data in the Decision-Support Process. In: *2015ProceedingsDecision Support Systems V – Big Data Analytics forDecision MakingBoris* . s.l.:s.n., pp. 10-21.

Van Vlasselaer, V., Bravo, C., Caelen, O., Eliassi-Rad, T., Akoglu, L., Snoeck, M. & Baesens, B. 2015, 'APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions', Decision Support Systems, vol. 75, pp. 38-48.

Wang, C. and Han, D., 2018. Credit card fraud forecasting model based on clustering analysis and integrated support vector machine. *Cluster Computing*, pp.1-6.

West, J. and Bhattacharya, M., 2016. Intelligent financial fraud detection: a comprehensive review. *Computers & security*, *57*, pp.47-66.