FINANCIAL FRAUD DETECTION USING MACHINE LEARNING

Thilini Upeksha Kodituwakku

(148225U)

Degree of Master of Science in Computer Science

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

March 2018

FINANCIAL FRAUD DETECTION USING MACHINE LEARNING

Thilini Upeksha Kodituwakku

(148225U)

Thesis submitted in partial fulfillment of the requirements for the degree Master of Science in Computer Science

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

March 2018

Declaration

I hereby declare that this is my own work and this thesis/dissertation does not incorporate any material previously submitted for a Degree or Diploma in any other University or institute of higher learning without proper acknowledgement and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date:

The above candidate has carried out research for the Masters/MPhil/PhD thesis/ Dissertation under my supervision.

Signature of the supervisor: Date:

Prof. Amal Shehan Perera

ABSTRACT

The method of performing transactions by means of payment cards is extremely efficient and the payment card industry is rapidly growing in popularity. However, the frauds associated with the payment cards are increasing and the patterns are evolving. Although a relatively smaller percentage is detected, fraud has become a major issue that affects the global banking industry. Machine learning techniques are widely used for payment card fraud detection.

The use of machine learning techniques generates successful results as there are large numbers of historical data that could be used for mining and manipulation. There are various machine learning algorithms available to construct fraud detection models. The main drawback of those models is their inability to deliver results accurately and efficiently at the level required by the industry as there is only a fine line between the fraudulent and non-fraudulent transactions.

The aim of this research is to create a model that reduces the present gap in the detection of payment card frauds using the ensemble machine learning technique. Ensemble methods are learning models that achieve performance by combining the opinions of multiple weaker models. The performance evaluation of a new ensemble model has been done on the real world financial data and the results indicated its capability of identifying a high percentage of frauds with low false alarm rate than the existing models in the payment card industry. Finally, results are analyzed, interpreted and directions for further research are suggested.

Acknowledgement

I would like to express my sincere gratitude to my supervisor, Prof. Amal Shehan Perera for providing me with the knowledge, support, and guidance throughout the duration of this project. Furthermore, I would like to thank my parents for the continuous support they gave me during this period. Also, I like to thank my colleagues who helped me immensely to gain an in-depth knowledge of payment card industry.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Description
ANN	Artificial Neural Networks
ATM	Automated Teller Machine
API	Application Programming Interface
EMV	Europay, MasterCard, Visa Standard
FDM	Fraud Detection Model
NFC	Near Field Communication
PIN	Personal Identification Number
POS	Point-Of-Sale
RFID	Radio-Frequency Identification
SMS	Short Message Service
SVC	Stored Value Card (also known as Prepaid Cards)

1. INTRODUCTION

In today's world, technological advancement allows daily tasks including banking and financial transactions to be accomplished with greater ease and comfort. When the payment cards such as Debit, Credit and SVC cards got introduced, people instantly started using them. The main advantage of these cards is, upon receiving them people could perform ATM, POS and E-Commerce transactions. Certain people use the payment card out of necessity or as an easy mode of carrying money. Today, this plastic card supports near-field communication (NFC), radio frequency identification (RFID) along with the magnetic stripe. The greatest revolution of the payment card came with the Chip and PIN technology also known as the EMV technology [1].

The logic behind the magnetic stripe card is that the magnetic stripe stores a unique secret number which represents the cardholder; anybody who claims that this number is known can take the ownership of the card. The major drawback is that cloning or skimming this card is rather easy and any person who claims to have this number can get the ownership of the card. Also, if somebody claims to know this secret number, he can get the ownership of the card [18]. This is a major drawback of the magnetic stripe.

As a solution for that three main companies which are in the business of transaction routing (EuroPay, MasterCard, Visa), developed an open-standard set of specifications for smart card payments and acceptance devices called EMV specifications. The EMV technology provides enhanced security features against fraud, such as data authentication, PIN entry, and cryptographic technology [18]. The added advantage of the EMV card is that the EMV chip embedded in the card provides the digital security even in offline mode.

Despite the recent security measures that have been introduced, payment card frauds are on the verge day by day. The fraud that is there today will be much more evolved tomorrow with the aid of technology.

Fraudsters are continually refining their methods, and as such, there is a requirement for detection methods to be able to evolve accordingly, which is one of the key challenges facing the financial services industry today.

Payment card fraud is a generic term used to describe a range of offences involving theft and the fraudulent use of payment card account data [18]. Payment card fraud can be committed in many ways. The fraudulent use of actual cards that are lost or stolen, obtaining cards through a fraudulent application process or the card that never arrives through the mail. Counterfeit is another fraud undertaken by altering the magnetic stripe fraudulently with illegally obtained account data. With the use of the internet as a channel for payments, card not present transactions became popular and at the same time frauds were committed using illegal payment card account data to undertake transactions. On the other hand, the errant merchant can always post transactions that never occurred from that customer. Therefore, the quicker the bank and the customer acts, the fewer frauds of this type can occur.

1.1 Problem Statement

As explained above, there occurs a large number of payment card frauds through the ATM and POS transactions. Almost all the cardholders are performing these two types of transactions regardless of the type of the card. It has been identified that credit card and debit card fraud resulted in total global losses amounting to $11.27 billion (5.2 Cents per US$100) during 2012, which is an increase of nearly 14.6% from the prior year [52][6]. Since it is hard to distinguish between the legitimate and illegitimate transactions of the bank, if unreported it goes without any investigation. Ultimately the customers of the bank unknowingly pay for the loss of income through higher fees or with high-interest charges. On the other hand, a large amount of funds is lost from the victim customer`s accounts due to fraudulent card payments.

1.2 Motivation

There is a huge financial loss because of the payment card frauds occurring through POS and ATM. The bank bears most of the up-front costs of these frauds while a portion of the liability can be passed on to the merchant and the customer. Moreover, the customers tend to lose the trust in the bank [53] [20]. Rapidly growing field of Information Technology not only increases the security of the transactions but also provides the ability to commit much more serious frauds.

Currently, there are systems which could identify these types of fraud, but the effectiveness is less than 70% [6] [12] [46]. It would be a great achievement if a model with a higher degree of fraud detection capabilities could be developed.

More effective fraud detection model which could capture a higher volume of payment card frauds may benefit all the parties concerned while discouraging activities of fraudsters. On the other hand, the reduction in frauds will result in a healthy financial sector which no doubt may provide a greater contribution to the economy as a whole.

1.3 Objective

The first objective of this research is to identify the attributes of the dataset that could be useful to build an effective fraud detection model (FDM). The success of the model partially depends upon the attributes that are selected at the initial stage. The next objective is to identify the appropriate machine learning technique for financial fraud detection. This is important since the identified machine learning technique will be used to build the model. Finally, the third objective is to build the fraud detection model using an identified machine learning technique that will result in a higher percentage of detection capabilities. The fraud detection model (FDM) should also be able to handle the skewed distribution and the changing variation of the card frauds.

2. LITERATURE REVIEW

The usage of financial cards is growing day by day due to its convenience. At the same time, the number of card-based financial frauds is also on the rise and evolving. [1] Payment cards worldwide experienced gross fraud losses of $11.27 billion in 2012, more than 14.6% over the previous year [6]. Payment card frauds are low risk and highly profitable criminal activities which bring a huge amount of easy money to organized crime groups can be the reason for an increase in numbers. These financial gains are invested in further developing criminal techniques or start legal businesses to generate further funds creating a never-ending chain of financial frauds.

There are many ways that payment card frauds can be done. It could be as simple as stealing the plastic card, counterfeit card or merchant posting transactions that never occurred. All these payment card frauds can be broken down into two categories [43]. Inner card fraud requires collusion between merchants and cardholders. External card fraud occurs when stolen, fake or counterfeit credit cards are used. The fraudsters can perform fraudulent transactions in two ways. Those are “card-present” transactions such as ATM, POS, etc., and “card not present” transactions such as the e-commerce transactions, which is not relevant to this study [13].

The payment cards play a critical part in the socio-economic development of any country [18] as many people at present tend to make financial transactions using payment cards. However, a proportionate growth of frauds is observed in those countries as well [18].Once a card is used to make a fraudulent transaction, the bank tends to re-issue the card which may cost a considerable amount of money when done on a large scale and also a huge amount of money is lost both to the bank and to the clients as the fraudulent transaction amount [20]. Even when the fraudulent transactions are identified the bank will make the clients pay for it as fees and interests in order to recover the amount lost [20]. Therefore, it is important to reduce the payment card frauds [3].

The above-mentioned reasons have made the field of payment card fraud detection very critical. There is a vast amount of literature available on this topic. In this section of the report, a literature review is done on the available literature with the intention of identifying the most appropriate methodology to proceed with the research under this topic.

The financial card fraud detection is closely related to machine learning because reasoning under uncertainty is a key element in artificial intelligence and particularly in machine learning [49]. Therefore, machine learning techniques have been increasingly used for financial card fraud detection and prevention. Many machine learning algorithms support reasoning under uncertainty, such as ANN, clustering, and genetic algorithm etc.

In the recent past, there has been a lot of research done on this topic. Therefore, it is important to conduct a thorough literature review of the existing work to identify the way to proceed with the "next step" in the financial card fraud detection and to identify the loopholes in the existing system.

In a nutshell, critical evaluation of previous literature has led to finding the following conditions that must be satisfied by a card based fraud detection model. These are described in detail in the section below [52].

1. The model should be able to handle skewed distributions.
2. The ability to handle noise.
3. Overlapping of data.
4. The model should be able to handle new kinds of frauds.
5. Identification of good metrics.
6. A number of transactions that can be processed in a second.

Although these models are successful to a certain degree, it is still extremely difficult to build an effective payment card fraud detection model due to various factors.

Firstly, there is a less amount of data available on the card-based transactions, and even if they are available all the parameters will not be available due to the security reasons [13]. Secondly, if the merchant commits a payment card fraud on the customer, it is extremely difficult to identify its legitimateness. Thirdly, the characteristics of the frauds change very often, so it is hard to fit into a certain model. Finally, the characteristics of the normal transactions are also rapidly changing making it very hard to identify [46].

The techniques that were used to build the financial fraud detection models can be divided into two main categories. Those are the supervised and unsupervised techniques. The supervised learning technique will make use of the previous data that were identified as fraudulent. The unsupervised learning will not categorize any data as legitimate or fraudulent.

2.1 Data mining

The data mining is a nontrivial extraction of implicit, previously unknown and potentially useful information [34]. In the process of knowledge discovery in databases, the knowledge, regularities, or specific information can be extracted from the relevant sets of data. So, nowadays large databases are very useful and reliable knowledge and information verification hubs in addition to the purpose for which they have been built. The discovered knowledge from those hubs could be used in different fields such as decision making in business, medical diagnosis, and in this case, the payment card fraud detection.

In data mining, the key task is the discovery of previously unknown knowledge. On the other hand, it is a computational process of discovering patterns in large datasets. There are two major approaches to data mining. Those are supervised and unsupervised as explained below.

2.2 Supervised learning

The supervised methods are classified with the class labels as a base to detect future transactions. The class labels in this case simply could be “legitimate” and “fraudulent”. The supervised classification methods can suffer from the imbalanced class labels. For example, the legitimate transactions are much higher than the fraudulent transactions for a given period. This will result in a skewed distribution. Supervised learning requires having very clearly defined classes. In these types of fraud detection sometimes it is unable to specify whether the transaction is legitimate or illegitimate [46] [50]. Therefore, it is quite impossible to have all the transactions in the sample to be clearly separated into two classes. The misclassification can harm the model. The supervised learning fails to identify the new types of frauds since the model has not come across the characteristics of the newer types of frauds.

Following are the classification approaches:

2.2.1 Classification

Classification is the method of supervised learning. That means the training data are accompanied by the labels indicating the class of observation. The new data are classified according to the knowledge gained from the training data.

Classification is a two-step process. The initial step is to build the model. Each record belongs to a pre-defined class. The model is built using the training data. To build the model the classification rules are also taken into consideration. The next step is the usage of the model. This step is to classify the future data that are coming into the model. The success of this step is measured by the accuracy of the classified data.

Popular classification techniques of fraud detection are as follows:

1. K-nearest neighbour classifier

2. Case-based reasoning
3. Rule-based approach
4. Fuzzy set approach

2.2.1.1 K-nearest neighbour classifier

All instances are represented by the data points in the n-D space. The nearest neighbour is determined using the Euclidean distance. The targeted function can be real-valued or discreet-valued. For the discrete-valued function, this classifier returns the nearest x-value. For the continuous-valued functions, the classifier returns the mean value of the K-nearest neighbours.

The K-nearest neighbour classifier is robust to noisy data, because of the averaging of the neighbours. The curse of the dimensionality, which means the distance, of two neighbours, could be depending on the irrelevant attributes

2.2.1.2 Case-based reasoning

The case-based reasoning is done by analyzing similar instances. Instances are represented by rich symbols. For example, graphs, charts etc., and multiple cases can be combined using this technique. Knowledge-based reasoning and problem-solving can be done through case-based reasoning.

The issues of using case-based reasoning are usually that the adapting to additional cases are hard and that the backtracking is hard.

Wheeler and Aitken [55] constructed a credit card fraud detection model using the case-based reasoning. This model functions by defining a set of features within a data or case base and then generating a similarity score that represents a relationship between a previously seen case and the test case. The authors have achieved 80% of

fraudulent transaction detection. The authors state that this model is very sensitive to smaller changes in the threshold.

2.2.1.3 Rule-based approach

Represent the knowledge in the form of IF-THEN rules. The rule covers an instance if it satisfies the conditions in the rule. The success of a rule can be measured by the accuracy and the coverage (fraction of records that satisfy the rule).

Conflict resolution is needed if more than one rule is triggered. This can be resolved by assigning priorities to each of the rules. Rule-based clustering can be done in two methods, namely direct method, and indirect method. In the direct method, which is also known as the Sequential Covering, the rules are extracted directly from data. In the indirect method, the rules are extracted from other classification methods. The typical direct algorithms are RIPPER, One R etc., and one of the widely used indirect method algorithms is C4.5 rules.

Sequential Covering follows the separate and conquer approach. The rules are learnt one at a time sequentially. Each time a rule is learnt, the records covered by that rule is removed. The algorithm repeats this process until the termination conditions are met. The stopping condition is applied when the rule is perfect, accuracy gets below a given threshold, or if the dataset cannot be split any further.

Advantages of rule-based classification are highly expressive, easy to interpret, easy to generate, can classify new instances rapidly, and can easily handle all types of values.

For credit card, fraud detection study has been done by Brause et al [5]., where the authors try to categorize all the fraudulent transactions as rules. For example, IF all the symbolic features are given THEN fraudulent transaction takes place. Combining several of these fraudulent rules together will result in a more generalized rule, and

by doing this, authors try to reduce the dependence of a rule on unimportant features. This model has achieved 95% of accuracy.

2.2.1.4 Fuzzy set approach

To measure the degree of membership, the fuzzy logic uses values between 0.0 and 1.0. The attribute values are converted to fuzzy values. For a given new sample, more than one fuzzy sample can be applied. Each rule contributes to a vote for membership into the relevant categories. The truth values for each predicted category are summed.

Lin et al [31]., have built a model for financial risk assessment using the fuzzy approach. The authors have used the key advantage of fuzzy logic to this model. That is, being able to describe the desired system behaviour using simple heuristics or IF-THEN rules. Since these rules are difficult to identify manually, the authors use the fuzzy clustering. Using this method, the user can specify the expected number of clusters or the system can find a likely number of clusters. The authors have achieved 78% of accuracy in this model.

2.2.1.5 Artificial Neural Network (ANN)

The Artificial Neural Network is a general statistical model with a large number of parameters. It repeatedly trains the training data with different weights as shown in the sample diagram in Figure 2.1.

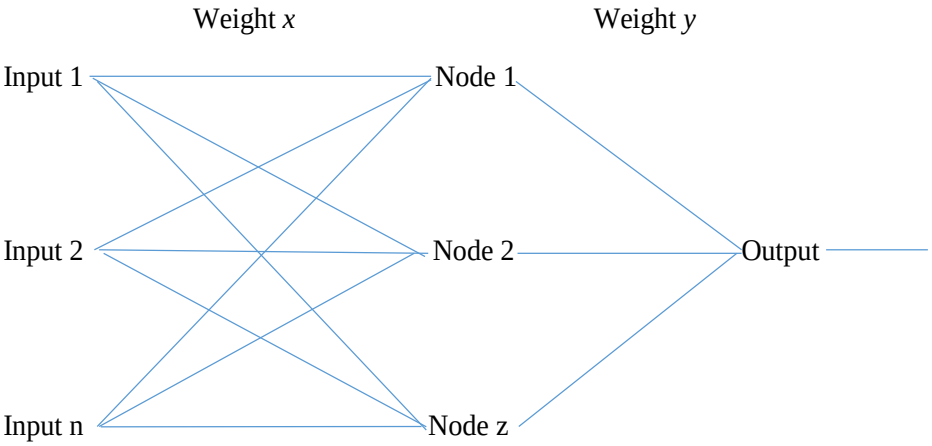

Figure 2.1: Diagram of Artificial neural network

The Artificial Neural Network has long been used and is very popular in financial fraud detection. In the design of neural network-based pattern recognition systems, there is always a process of feature extraction that is applied to the input "raw" data in terms of the particular pattern recognition problem at hand. Neural networks can calculate the profiles independently, giving the capability to analyze the behaviour of the users more closely [14].

There are two phases in ANN, training and recognition. The learning is referred to as training in ANN and could be categorized into two types. As explained above they are supervised and unsupervised learning. ANN yields best results when applied to large datasets. Hence, making it very appropriate to payment card fraud detection as the latter is accumulating a huge volume of transactions.

Sushmito Ghosh and Douglas L. Reilly [44] developed such a model using the artificial neural network. The neural network used in this research was P-RCE Neural network, which is a type of a radial basis function network used for pattern recognition. P-RCE consists of a three-layer, feed-forward network that is highlighted by its capability of making only two training passes through the training dataset. The initial round involves a process of a prototype where the samples of the training set are stored in the weights between the first and the second layers of the network. The final round of the training determines local posteriori probabilities associated with each of these prototype cells. The accuracy obtained using this model

was 20-40% reduction of total fraud losses. The objective of this study was to determine if a wide range of information is used to characterize a transaction, which can be helpful in developing an improved fraud detection capability. What was envisioned is a system that could review each transaction in the context of the recent history of its account and the pertaining payments, along with other non-financial data of the transaction, to determine the likelihood of fraud. Similar work using neural network was done by Cardwatch (Aleskerov et al., 1997) [11].

2.2.1.6 Bayesian network

The Bayesian network has been widely used in building fraud detection models. A Bayesian network is a directed acyclic graph which consists of random variables [40]. Each node of the graph is represented by the variables of the selected domain. Relationship of the nodes and the edges represent the probabilistic dependencies. Two nodes are said to be conditionally independent if there are no edges connected between them. Each node of the network has a conditional probability table that quantifies the effect of the parent node.

Bayesian networks are very suitable when certain information is known, and the incoming transaction is undecided. When it comes to the payment card fraud detection, the Bayesian belief network will give the probability of the transaction being fraudulent [2].

According to S.Maes et al [46], two models are built using the Bayesian networks to detect credit card fraud. The first model is built to capture the behaviour, assuming that the customer is fraudulent, and the second is built assuming the customer is legitimate. The authors further use the Bayesian model to: 1. To identify the topology of the network, especially the missing links and the direction of the arrows, 2. Learn the numerical parameters for a given network topology.

This model has achieved 73% accuracy, 15% of the transactions are incorrectly classified. The authors point out that although the training period is shorter; the actual fraud detection process is relatively slow.

2.2.1.7 Decision trees

Decision trees are a method to present data in a hierarchical, sequential structure. A decision tree can divide a complex problem into many subproblems [8]. The tree will start from the root and each node of the tree is split into binary fashion or in the multi-split. There are two phases when doing fraud detection through decision trees. First, is to generate the decision tree, and the second is to apply the decision rules to determine the class of the incoming transaction [43]. The tree will match the decision rules for the particular transaction. Then according to the matching criteria, the transaction will be assigned a class. If the matching criterion does not indicate a class clearly, it will determine the class with the highest risk and the transaction is assigned to that class [43]. ID3 and the C4.5 are the most widely used algorithms of decision trees.

Aihua Shen et al. [45], performed a comparison between ID3, Neural networks, and logistic regression. The authors have found that 38.94% accuracy is gained using the ID3 algorithm.

2.2.1.8 Genetic algorithm

Although the Genetic algorithm is mentioned as a classification technique [56], this is used for optimization. The Genetic algorithms are also used for predictive methodologies. A genetic algorithm is inspired by natural evolution. The fundamental idea behind this algorithm lies behind the Darwinian evolution, where the species undergo a selection process and from that, only the fittest will survive. In the Genetic algorithm, the solution to the issue at hand can be considered as an individual and the measure of fitness is evaluated through the objective function. The other solutions are considered as the other species in which the current solution should compete with. Therefore, the Genetic algorithm works with the entire population of solutions. This yields good results in fraud detection [35]. The major difference between the Genetic algorithm and the Evolutionary strategies is that the

Genetic algorithm relies on the cross-over, a mechanism of probabilistic and useful exchange of information among solutions to locate better solutions [8].

There are three main steps of Genetic Algorithm [35]

Selection – The fitter solutions survive, while the weaker solutions will perish. The fitter solutions will receive a higher number of offspring with the higher chance of surviving the generation and the weak solutions will receive less than one solution.

Crossover – The solutions are picked randomly from the population to be subjected to the crossover.

Mutation – The resultant solution after the crossover is subjected to mutation. The mutations will occur independently.

The fitness function [56] transforms the measure of performance into an allocation of reproductive opportunities. This evaluation of each member of the population is known as a chromosome representing a set of parameters which is independent of any other chromosome.

E.Duman and H. Ozcelik [12] has built a model with the aid of the Genetic Algorithm combined with the Scatter Search. The objective of this research was to build a model that identifies credit card fraud by customer behaviour module. Many behavioural variables are evaluated, and if a value of a variable is larger than the average plus a specified number of deviations, a certain suspiciousness point is generated. Following is the gist of the research [12]:

The authors have identified 43 parameters from the card base that are important. One of the 43 parameters is picked up and changes its value randomly within its available range. Then the children are generated using the recombination operator. One of these children are randomly selected and the mutation operation is applied. So, the fitness of this individual solution is determined by the total amount of savings gained from the total identified frauds. The three mutations with the highest value are

selected and transferred to the next generation. This procedure was continued until no improvements are made for the next 10 cycles.

The results were recorded in terms of the savings that was incurred by the identified frauds. The savings value threshold was 40%, the authors claim that this generated about 150% of false positive alerts which is a drawback.

2.3 Unsupervised learning

The unsupervised methods occur where there are no class labels. It can be used to identify outliers or the groups to which each transaction belongs. It will not assume that the fraudulent transactions would have the same characteristics as the current [53] [24]. Unsupervised learning has the ability to adapt to the new types of frauds. But the disadvantage is that it will assign a very high suspicion that is anything out of the defined boundary.

The approaches such as the neural networks can have the ability to be used in both methods [15].

Clustering is an unsupervised data mining method. Following are the approaches:

2.3.1 Clustering

Cluster analysis is the organization of the collection of the patterns into a cluster based on the similarity [3]. In clustering, a collection of unlabeled data is given to cluster into meaningful groups. Clustering techniques are used primarily for behavioral model analysis. In many cases, there is little prior knowledge available on the data. Therefore, the user is compelled to make a few assumptions. However, cluster analysis can suffer from a bad choice of metric selection.

The clustering task comprises of the following steps [3]:

1. Pattern representation
2. Definition of pattern proximity measure appropriate to the data domain
3. Clustering
4. Data abstraction
5. Assessment of output

Bolton & Hand (2002) [40] have conducted research on behavioural model analysis using clustering techniques. It detects individual objects with the use of the parameters. In financial fraud detection, the current behaviour is compared with the previous behaviour of that object. The objects which have a significant deviation from the previous behaviour are given a close investigation of the behaviour. Similar work has been done by E.Kirkos et al., (2007) [15], T.Fawcett and F.Provost (1996) [49].

Following are the mostly used clustering methods:

1. Partitioning method
2. Hierarchical method
3. Density-based method
4. Conceptual method

2.3.1.1 Partitioning method

If there are x number of data points, there will be k number of partitions from the data. Each of those partitions will create a cluster. Therefore, there will be k number of clusters. Each cluster contains at least one data point. Each data point belongs to exactly one cluster.

The partition method will create an initial clustering. Then it will use the relocation technique to improve the other clusters. K-means clustering is the most known algorithm.

Marathe and Shawky [4], performed a study on investment funds fraud detection study using the K-means clustering. The authors apply K-means clustering to categorize mutual funds. The clusters are assigned based on the customers` investment objectives and compared with the expectations and the financial characteristics. In the study, the authors found that 43% of the funds are misrepresented.

2.3.1.2 Hierarchical method

A hierarchy is developed for the given set of data. There are two techniques used. Those are the agglomerative method and the divisive method.

The Agglomerative method is a bottom-up approach. It starts with each data point creating a group. It merges the groups that are close to one another. It keeps on merging until the termination conditions are applied. The Divisive approach is a top-down approach. Here the algorithm starts with the data points of the same cluster. It continuously splits the clusters into smaller clusters. This process happens iteratively until the termination conditions are met.

In order to detect telecommunication fraud, a model was built using the hierarchical clustering [21], where the user does not specify the number of expected clusters, k. The algorithm creates a tree-like hierarchical structure where all the values of, k, is contained. The root of the tree defines all the clusters in the hierarchy. The leaves of the hierarchy contain the objects. The agglomerative clustering starts with each of the node and proceeds by combining the closest nodes until a cluster is obtained.

The authors of this study obtained 80% of TP rate and a 2% of FP rate. However, they claim that not being able to get a definition of the discriminating characteristics was a disadvantage.

2.3.1.3 Density-based method

This method uses the data density for clustering. The algorithm continues to grow the cluster as long as the neighbouring density increases with a certain threshold. The radius of a cluster should contain the least minimum amount of data points.

2.3.1.4 Conceptual method

Conceptual clustering is a technique that forms concepts out of data incrementally by subdividing groups into subclasses iteratively, thus building a hierarchy of concepts. It forms a tree, as root creates new children using attributes. Each node is a class. The algorithm generates a concept description for each class. Two major algorithms used for conceptual clustering are Cluster/2 and Cobweb.

Cluster/2 is one of the initial conceptual clustering algorithms. The algorithm works as a meta-learning scheme. The cluster/2 algorithm creates x categories by constructing individual data points grouped around x-seed objects. The Cluster/2 algorithm is obsolete, but it introduces the concepts for current algorithms.

Cobweb is an incremental conceptual clustering algorithm, which builds a hierarchy of clusters without having a pre-defined number of clusters. The clusters are probabilistically represented using conditional probability. $P(Y=x|C)$, with attribute Y, as value, x, given that the instance belongs to class C. The algorithm starts with an empty root. For that, the instances are added gradually. This algorithm searches the possible hierarchies, an evaluation function based on the category utility. Category utility is the intra-class similarity and the inter-class similarity for each of the individual data point.

In support of detecting revenue fraud, a patented fraud module using the conceptual clustering was built to recognize the patterns from the significant events [17]. The conceptual clustering will be used for generating a collection of classes based on the

historical data. This method is useful for detecting the incidence of fraudulent activity from very large amounts of data in terms of tax returns or insurance claims.

2.4 Comparison of approaches

A summary of the approaches can be drawn using the above literature review [2] [3] [8] [11] [14] [35] [40] [43] [44] . The metrics that were considered are the accuracy and the speed of fraud detection. Table 2.1: shows the results of the comparison adopted from Zareapoor, M. et al [61].

Table 2.1: Comparison of approaches

Method	Accuracy	Speed of fraud detection
Artificial neural network	Medium	Fast
Bayesian network	High	The training period is faster, but fraud detection speed is moderate.
Genetic algorithm	Medium	Fast
Clustering	Medium	Fast
Decision trees	Medium	Fast

2.5 Ensemble methods

This method uses multiple models combined to generate the output. The ensemble algorithm combinations can be made based on the interest of the domain. It is known to generate better results than single algorithms. There are three types of reasons for the ensemble method algorithms to perform better, namely statistical, representational, and computational [25].

Statistical – If the learning algorithm is assumed to search a space, M, to find the best match, without adequate test data. The algorithm will make various hypothesis using the insufficient amount of data on M, and all of them will generate the output with

almost the same accuracy on the training data. In this case, using the ensemble method will average the results of its member algorithms` hypotheses to avoid making mistakes.

Representational – In a space, M, with the common limitations, it is hard to find the exact function, f. If a set of classifiers are combined to create an ensemble it is possible to have a number of functions to represent the space, M. Although the neural network also has the capability of representing several functions, it will halt when the model fits the training data.

Computational - If the learning algorithm is assumed to search a space, M, to find the best match, it can catch up in the local optima. Therefore, commencing at various points and combining the models into an ensemble, will be closer to the exact hypothesis.

Some of the algorithms that can be used to build ensemble models are discussed below.

2.5.1 Bagging

Bagging, also known as the “Bootstrap Aggregating” is done by generating many similar copies of the original training example by sampling with replacement [28]. For a given bootstrap sample, an example in the training set has probability, $(1 - 1/\mu)^{\mu}$ of being selected at least once in the μ times. When μ is large, the limit as $\mu \rightarrow \infty$ the formula is approximately equal to $1 - 1/e = 63.2\%$. Therefore, when drawing μ items with replacement from a large sample of size μ, on average the sample contains 63.2% of the original observations and omits 36.8%.

Approximately 63.2% is taken for bootstrap sampling from the original examples [26]. The classifiers will run on each bootstrap, and combine the output, usually using the majority voting to create the final output. Bagging minimizes the noise, bias, and

variance by averaging over bootstrap samples. Using the bagging method, models with high accuracy could be created.

The Bagging algorithm is given in Figure 2.2.

BAGGING

Training Phase

1. Initialize the parameters
   - $M = \emptyset$, the ensemble,
   - L = the number of classifiers to train.
2. For $k = 1, \ldots, L$
   - Take a bootstrap sample S_k from Z.
   - Build a classifier M_k using S_k as the training set.
   - Add the classifier to the current ensemble, $M = M \cup M_k$.
3. Return M.

Classification phase

4. Run $M_1, \ldots, M_L$ on the input x.
5. The class with the maximum number of votes is chosen as a label for x.

Figure 2.2: Bagging algorithm

2.5.2 Boosting

Boosting is following the concept of assigning weights for the training samples. The weights are redistributed after every prediction. Weights of the correctly classified examples will increase while the weights of the misclassified examples will decrease [41]. The most popular boosting algorithm is the AdaBoost algorithm which is also known as “Adaptive Boosting”.

In the AdaBoost algorithm, the assigning of the weights is done as above, but the weak classifiers are focusing more on the difficult examples in the dataset. It creates

more classifiers which focuses on the difficult areas. The final output is determined by the weighted voting [26].

2.5.3 Arcing

Arcing is also known as “Adaptive Re-weighting and Combining”. Sampling examples will be replaced by the original examples probabilistically [29]. The probability of each example is calculated by determining how frequently it is being misclassified previously. As Boosting, Arcing makes the classifiers learn from difficult areas.

2.5.4 Random trees

Random trees, which were originally known as “Randomized trees”, chooses a variety of combination of attributes that form the entire set of attributes. For each of the attribute combination, this model trains with the decision tree to build the best tree with various heuristics [58]. The number of times the training dataset will be scanned is equal to the number of combinations of attributes. The output is generated by the arithmetic averaging of all output from all trees. This method is most suitable for samples not seen in the training dataset.

The classical C4.5 algorithm can be modified to be a random tree by injecting randomness into the attributes selection at each node of the tree [26]. In each node, the random C4.5 decision tree selects an attribute randomly out of the best twenty attributes [58]. The randomness helps to pick up the attributes that have not been selected before.

2.5.5 Random forests

The random forests differ from random trees by not limiting to the best twenty attributes [30]. Bagging is also used in random trees to produce training sets to different trees. The CART is one of the popular random forest algorithms. In CART algorithm, the output is generated by using majority voting technique [28].

In random forests, the accuracy depends on the selected classifiers and the dependency between them. The algorithm is shown in Figure 2.3.

Input: D = $\{(x_1, y_1), \ldots, (x_m, y_m)\}$

for b = 1…*B* do

Z ← set of *size N* drawn from *D* by sampling with repetitions

Grow a tree T_b by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.

i. Select *m* variables at random from the *p* variables.
ii. Pick the best variable among the *m*.
iii. Split the node into two children nodes.

$T_B = T_B \cup T_b$

end for

Output: ensemble of trees T_B

Figure 2.3: Random forest algorithm

2.6 Attribute selection

The limited amount of committed transactions available makes it difficult to do research on this topic [44] [13]. This is due to the sensitive financial data that is kept confidential for reasons of customer privacy. In a single committed transaction, the data like the transaction amount, date, time, destination, and the merchant category code (MCC) are the most important attributes that can be extracted [10]. However, the results with the real data for real instances are not published in the previous literature[46]. These attributes are used to capture the information as much as about the transactions, this will be used in the model to detect the fraudulent transactions, as explained below.

Scenario 1 – The customer has done two different transactions in two remote destinations within a very short period of time. To detect this kind of scenario, the attributes to be used can be card number, amount, transaction date/time, and location that are already being stored in the database.

Scenario 2 – If the system has detected an unusual amount of transactions within a short period of time, the algorithm can recognize this as a fraudulent transaction using the particular card number, amount, and transaction date/time.

Scenario 3 – The system can detect the high risked countries using the list maintained in the database, comparing the merchant category code, currency code, and the country code.

Scenario 4 – The algorithm can detect the transaction count and amount to identify the pattern of daily/monthly consumer expenditure amount/count. If the transaction amount exceeds this, it can be a fraudulent transaction. The attributes to identify such transactions are card number, account number, location, transaction amount and transaction date/time.

Scenario 5 – If transactions are coming from a suspended, deleted, or expired card, this can be detected using card expiry date, and processing code.

Scenario 6 –The system can flag and maintain the customers who had not settled the credit card amount at the proper due date and this can be used to identify the blacklisted customers. These customers can be identified using the card number, and account number.

Scenario 7 – In an advanced scenario where the transactions are coming from EMV and magnetic stripe cards, it can be correctly recognized by the POS entry mode attribute. If the POS entry mode is incorrect for the relevant card type, means it is a fraudulent transaction.

2.7 Challenges

2.7.1 Skewed distribution

The fraud detection model should be able to handle skewed distribution since only a very small percentage of all transactions are fraudulent. When the model is trained with the labelled data samples there is a possibility that one of the classes (class A) gets a considerably higher amount of data than the other class (class B). This situation will cause skewed distribution because the model knows very little about class B. In payment card fraud detection, the skewed distribution is a common problem since the number of legitimate transactions are much higher in number than the fraudulent transactions. Dividing the training sets into pieces to make the distribution less skewed is a solution to this problem [13] [44] [46].

2.7.2 Handling noise

The dataset can have erroneous data or sparse data, referred to as noisy data. Noise is an unavoidable problem which affects the data collection and data preparation processes in Data Mining applications, where errors commonly occur. Noisy data

may affect the intrinsic characteristics and may show unwanted new characteristics in the problem domain, as an example, noise can lead to the creation of small clusters or disappearance of specific classes [38]. The datasets from the real world often suffer from the noisy data.

There are two types of noise.

1. Class noise – Occurs when the classes are incorrectly labelled. This can occur during the circumstances of the labelling process, inadequate data, or the data entry errors.

2. Attribute noise – Corruption in the attribute values. This can be because of the incomplete attribute values, missing values, or erroneous attribute value.

To avoid noise, data should be cleaned before processing [32]. There are two methods how this could be done [38].

1. Design the algorithm to handle the noisy data

2. Remove the noisy data during the pre-processing

The latter method is more popular for handling noisy data. The algorithm design must incorporate the required features without making adverse effects to its other capabilities.

2.7.3 Overlapping of data

Overlapping of data occurs when the legitimate transactions have the characteristics of the fraudulent transactions or the fraudulent transactions have the characteristics of the legitimate transactions [53]. The partition between the fraudulent and legitimate transactions should be mutually exclusive. It is important that the fraud detection model takes the correct decision so that the fraudulent transactions will not pass through as legitimate since it may lose more money than blocking a legitimate transaction identified as a fraudulent transaction.

2.7.4 Handling new types of frauds (Concept Drift)

The system should be able to handle new kinds of frauds since various kinds of new frauds are popping up very often. Fraudulent activities will be dynamic and continuously change the pattern before it becomes common knowledge. Therefore, adapting to detect new kinds of frauds should be a feature of a good card fraud detection model [46].

2.7.5 Identification of good metrics

While the financial sector is falling victim to fraudulent transactions, this comprehensive review found that most of the existing fraud detection models failed to identify fraudulent cases efficiently with a reasonable precision. Nevertheless, the study led to finding strengths and weaknesses of many existing models which could be helpful when building new models.

Identification of good metrics is very important as the efficiency and the accuracy of the entire model will be depending on it. If the metrics are not carefully chosen the accuracy level of the model may be stated as high but with a very high degree of misclassifications of data [8] [13] [53].

Given that the dataset is labelled, it is best to use supervised learning. Out of all the supervised learning methods, the ANN has been the most widely used fraud detection algorithm, because of its ability to handle large datasets. Furthermore, the literature review has been providing strong evidence that Ensemble models, where a group of base models is combined, generate better results than a single algorithm. To run these models successfully it is important to identify quality attributes, which will capture useful information of a transaction. A good payment card fraud detection model should be able to handle the skewed distribution, noise, and the new types of frauds.

3. METHODOLOGY

3.1 Objective

The objective of this research is to develop a highly accurate and efficient fraud detection model (FDM). Therefore, the entire research has been carried out focusing on its accuracy and the efficiency of the rate of capturing the fraudulent transactions. During the project period, extensive studies have been carried out;

1. Finding the most appropriate combination of algorithms which perform well in this task.
2. Identifying effective parameters that could be used to build the fraud detection model.
3. Identifying a model which is more efficient than the currently available models.

3.2 Approach

A payment card fraud detection model (FDM) has been developed to address the above-mentioned issues with the following characteristics.

1. To handle the ever-changing patterns of the frauds
2. To perform fraud detection very efficiently

A payment card fraud detection model's main task is to capture a high amount of frauds. Therefore, it should have a very high adaptability to the changing fraud patterns. To be integrated with other modules (such as a payment switch), the fraud detection model must be highly efficient, since a transaction in a normal scenario is completed within 8-10 seconds and because slow detection leads to higher losses.

By selecting a set of parameters for the fraud detection model, that most of the financial software uses when performing a transaction, the model is expected to be easily coupled with the industry available ATM/POS financial solutions.

The fraud detection models that are existing in the banking and financial sector are generally rule-based models where to enter the rules to the system, there needs to be human intervention. If a transaction satisfies this rule, it is classified as fraudulent. This may not always be the case. There are transactions, although they seem to have genuine parameters from the surface level, that can be legitimate or vice versa. The rule-based classification cannot completely be ignored as when the service mandates are issued from the authoritative networks (eg. VISA or MasterCard), they specify a set of rules by which the transactions should be monitored. Therefore, while the model has the capability of accommodating those rules, it should also be a much-extended version to accurately identify the changing fraudulent patterns. This model is also inspired by the work of Patent US5822741 - neural network/conceptual clustering fraud detection architecture, by S. Fischthal and L. M. Corporation. (1996, Feb. 5). [Online]. Available: http://www.google.com/patents/US5822741.[39], where the pertaining model was created for insurance fraud detection but lacks the proper utilization of the rule-based classification. This model is built to gain knowledge to identify changing fraudulent patterns and to meet the current needs of the market.

3.3 Scope

The scope of this fraud detection model (FDM) is to carry out the successful identification of the fraud transactions efficiently and with a high degree of accuracy.

It is a hybrid model of the classification and the neural networks. Therefore, in the classification phase, the following two processes will be carried out;

1. The most appropriate attributes will be chosen
2. Carry out the classification

After the data is classified using the specified rules and according to the behaviour of the attributes, the rule-based algorithm will run to identify most likely suspicious or fraudulent transactions based upon the specified rules. The bagging ensemble method

with the Levenberg-Marquardt algorithm is there for further identification of fraudulent transactions.

3.4 Data selection

At the initial stage, a very large amount of historical data was involved. This historical data was stored in a DBMS (ideally an operational database since this tends for live operation but could be a Data Warehouse as well). For this implementation, an operational database was used, but the system can be extended to retrieve data in XML, XLS or CSV formats.

The historical data that was received has known characteristics. Each transaction in this set of databases was included with a large set of attributes. Sometimes some of these fields can be NULL or empty. In order to retrieve diversity of patterns of transactions, the historical data to build this model was not extracted from a single database instance. Therefore, there are records of multiple database systems (such as PostgreSQL, DB2, and Oracle). For the task of processing these transactions, the transactions from PostgreSQL and DB2 were migrated to the Oracle SQL.

From the migrated database, the most suitable attributes were selected using the expert knowledge, such as the PAN, transaction amount, expiry date, institute id, authorization code, MCC, terminal id, added date etc., These attributes are stored in memory and could be used when new transactions come in.

3.4.1 Dataset

In order to evaluate the performance of the FDM model, three large datasets containing a total of 5,700,000 recent payment card transactions from three financial institutions (which includes data of one offshore bank as well) have been obtained. It contains card transactions spanning over the period of the last three years. The set of data are labelled, and it includes approximately 3% of fraudulent transactions.

3.5 Data pre-processing

When it comes to transaction database, incomplete fields are common, since all the attributes are not important for a particular type of transaction or simply it could be an incomplete transaction. If the transactions are coming through a network, the unwanted fields and the blank values of the transaction are filled with N/A values. To handle these values the package "***norm***" [23] in R, was used. ***Norm*** is an open-source, BSD licensed library providing high performance, easy to use data structures. In the BSD package, a range of features is available for missing data analysis.

The da.norm function which is included in the norm package is used for data augmentation of incomplete multivariate normal data. [54] This function is capable of simulating a single or multiple iterations of a single Markov chain. At each iteration, the missing values are randomly imputed, given the observed data and the current parameter value. [41]

3.6 Splitting the data

The datasets were split into two parts:

1. Data to train the classifier
2. Data to test the model

A set of 2 million records was used as training data. These records were randomly selected through permutations.

3.7 Training data

The samples of attributes used for each of the transactions are given in Table 3.1. These parameters were selected upon receiving expert advice, furthermore, they are mandatory to complete a transaction. The descriptions of the following parameters are included in the MasterCard, "IPM Clearing Formats," 2015.[33]

Table 3.1: Attributes used in transactions

Attribute Name	Data Type	Data Description
Primary Account Number (PAN)	VARCHAR2 (40 BYTE)	A series of digits that identify a customer account or relationship.
Amount, Transaction	NUMBER(19,0)	The amount of funds the cardholder requested in the currency appearing on the transaction information, which may be the acquirer's local currency or a currency acceptable to the cardholder and card acceptor.
Amount, Cardholder Billing	NUMBER(19,0)	The amount converted to the issuer's designated cardholder billing currency.
Date, Expiration	DATE	Specifies the year and month after which a card expires.
Point of Service Data Code	VARCHAR2 (4 BYTE)	(POS) is a series of codes that identify terminal capability, terminal environment, and point-of-interaction security data.
Card Acceptor Business Code (MCC)	VARCHAR2 (4 BYTE)	Classifies the type of business applicable to the card acceptor.
Acquiring Institution ID Code	VARCHAR2 (11 BYTE)	Identifies a transaction acquirer.
Forwarding Institution ID Code	VARCHAR2 (11 BYTE)	Identifies a message's forwarding institution.
Retrieval Reference Number	VARCHAR2 (24 BYTE)	Retains the transaction's original source information.
Card Acceptor Terminal ID	VARCHAR2 (15 BYTE)	A unique code identifying a terminal at the card acceptor location.
Card Acceptor Name/Location	VARCHAR2 (40 BYTE)	Contains the card acceptor's name and location as known to the cardholder.
Date and Time, Local Transaction	DATE	Refers to the local year, month, day, and time at which the transaction takes place at the card acceptor location.
Processing Code	VARCHAR2 (6 BYTE)	Refers to a series of digits that describe the effect of a transaction on a customer account and the type of accounts affected.
Currency Code, Transaction	VARCHAR2 (3 BYTE)	Defines the transaction currency.

As the initial step, the historical data were classified according to the rules that were specified. After it had been assigned to classes, the transactions become input to the neural network. The output of this was taken for analysis. Then the set of transactions that was set aside to act as the new transactions were sent to the payment switch as in a real transactions flow. The rule can be specified whether to block a transaction or to allow the transaction to proceed with a warning to the customer. Obtaining good results from this model will also depend on selecting a good set of attributes. If an important attribute (e.g.: Card Acceptor Name/Location) is failed to add to the list of attributes, the features that were utilized to identify the transaction will be missed and may have an adverse impact on the effectiveness of the output.

The classes were derived from the rules that were specified by the users (of the fraud detection model). This is important since the rules can be added according to the changing patterns of fraud. Furthermore, it gives a reason on why the particular transaction was categorized as a fraudulent transaction, rather than in a plain statistical model which may not provide such feedback to the user. The neural network is performing more efficiently because the transactions were already classified into the complementary subsets by the ten-fold cross-validation technique.

It is important to have the neural network (Levenberg-Marquardt algorithm with bagging) because relying solely on the user-specified rules will only provide an initial filtration. The neural network is employed to minimize the possible two errors, false positives (type 1 error) or false negatives (type 2 error) in the results.

3.8 Levenberg – Marquardt algorithm

The neural network is a very popular data mining technique. It has a powerful and a generalized framework to represent non-linear mappings. From the types of the neural networks, the multi-layer perceptron and radial-based function neural networks are most commonly used for mining massive amounts of data.

The error backpropagation is used in the search for derivatives of an error function with regards to the parameters of a multilayer perceptron neural network. The mean squared error is the commonly used function. Gradient descent function is the simplest optimization method that can be used. These are also known as the “first-order optimization algorithms”.

3.8.1 First-order optimization algorithms

Backpropagation algorithm with the gradient-descent is the basic first-order optimization algorithm. The gradient descent of the convergence is low because it tends to have a zig-zag behaviour on the flat error surface. This method ignores the minor features in the error surface and does not get stuck in the local minima. The gradient descent is sensitive to the learning rate. If the learning rate is too high the performance can become unstable. If the learning rate is small, it will take a long time to converge. To increase the performance of the gradient descent, it should try to keep the adaptive learning rate high while keeping it stable.

A resilient backpropagation algorithm can be used to optimize the convergence of the gradient descent. In this case, the parameter update is determined using the sign of the derivative by the resilient backpropagation algorithm. Therefore, if a change in a parameter takes place in the same direction during several iterations, the rate of the parameter change is high. When the parameter change oscillates, the rate of the change of parameter is reduced. Although it is less accurate, the resilient backpropagation increases the rate of training in flat regions of the error function.

3.8.2 Second-order optimization algorithms

The error function reduces while moving towards the gradient, although this does not guarantee the optimal convergence. Therefore, to find the optimal convergence, the

second-order optimization algorithms use the gradient and the Hessian matrix. The concept of searching for the global minima of the second-order optimization algorithms is same as the first-order algorithms but by using an improved function.

The conjugate algorithms are much faster than that of gradient descent with an adaptive learning rate. The search function of a conjugate gradient is performed towards the direction of the gradient which will reduce the performance function. A scaled conjugate function will further reduce the rate of training since it will avoid the line search. The memory needed for the conjugate gradient function is a little more than that of the gradient descent.

An alternative to the conjugate gradient function is Newton`s method for the fast optimization since the convergence is faster. The drawback of using Newton`s method is that the computation of the Hessian matrix is expensive to compute, and there is no conformity of the multi-layer perceptron neural networks of being singular or non-singular. Then there are Quasi-Newton methods which are similar to Newton`s method with the line search. The difference between Newton`s method and the Quasi-Newton method is that the latter uses a symmetric positive definite matrix which updated at each cycle, approximates to Hessian matrix. This needs more computational capabilities than the above methods.

When the objective function is a sum of squares and the Hessian matrix is a linear approximation of residuals Newton`s method becomes the Gauss-Newton method. In this method, the second derivative matrix can be approximated using the information required to determine the first derivative vector. Although the convergence of the Gauss-Newton method is more than that of the Quasi-Newton method, it has a large residual issue. If this issue arises during the training, the convergence rate will reduce rapidly. This issue can be overcome by using the Levenberg-Marquardt algorithm.

The pseudocode of the Levenberg-Marquardt algorithm [32] is given in Figure 3.1

Input: A vector function $f : R^m \rightarrow R^n$ with $n \geq m$, a measurement vector $x \in R^n$ and an initial parameter estimate $\mathbf{P}_0 \in R^m$

Output: A vector $\mathbf{P}^+ \in R^m$ minimizing $||x - f(\mathbf{P})||^2$.

Algorithm:

$k := 0$; $v := 2$; $\mathbf{P} := \mathbf{P}_0$;

$\mathbf{A} := \mathbf{J}^T \mathbf{J}$; $\epsilon_\mathbf{p} := x - f(\mathbf{P})$; $\mathbf{g} := \mathbf{J}^T \epsilon_\mathbf{p}$;

stop := $(||\mathbf{g}||_\infty \leq \epsilon_1)$; $\mu := \tau * \max_{i = 1,\ldots,m} (A_{ii})$;

while (not stop) and $(k < k_{max})$

    $k := k+1$;

    repeat

        Solve $(A+\mu\mathbf{I})\delta_\mathbf{p} = \mathbf{g}$;

        if $(||\delta_\mathbf{p}|| \leq \epsilon_2 ||\mathbf{P}||)$

            stop: = true;

        else

            $\mathbf{P}_{new} := \mathbf{P} + \delta_\mathbf{p}$;

            $\rho := (||\epsilon_\mathbf{p}||^2 - ||x - f(\mathbf{p}_{new})||^2) / (\delta_\mathbf{p}^T (\mu\delta_\mathbf{p} + \mathbf{g}))$;

            if $\rho > 0$

              $\mathbf{P} = \mathbf{P}_{new}$;

              $\mathbf{A} := \mathbf{J}^T \mathbf{J}$; $\epsilon_\mathbf{p} := \mathbf{x} - f(\mathbf{P})$; $\mathbf{g} := \mathbf{J}^T \epsilon_\mathbf{p}$;

              stop := $(||\mathbf{g}||_\infty \leq \epsilon_1)$ or $(||\epsilon_p||^2 \leq \epsilon_3)$;

              $\mu := \mu * \max(⅓, 1 - (2\rho - 1)^3)$ $v := 2$;

            else

              $\mu := \mu * v$; $v := 2 * v$;

            endif

        endif

    until $(\rho > 0)$ or (stop)

endwhile

$P^+ := P$;

Figure 3.1: Pseudocode of the Levenberg-Marquardt algorithm

The Levenberg-Marquardt is a restricted step (step size can be determined using the slope of the tangent or using the difference between the predicted and the actual) algorithm, therefore the large residual issue could be avoided. Due to its rapid convergence and robustness, the Levenberg-Marquardt is the standard method for solving non-linear squared problems. The Levenberg-Marquardt method starts with a small step size and transforms more like Newton`s method using the Hessian matrix near error minimum. This increases the rapid convergence and high accuracy. Therefore, where fastest and accurate results are expected, the Levenberg-Marquardt method has been widely used.

3.9 Implementation and Methodology

In the previous chapters, the algorithms, their outcomes, and the usages were discussed. The motive of the literature review was to determine the most suitable algorithms to be used in the fraud detection model (FDM). In this section, the methodology of the proposed ensemble FDM model (shown in Figure 3.3) will be discussed.

3.9.1 User Entered Rules

There are three ways that the users of this model (typically a financial organization) would know the basic rules that have to be entered to detect a basic fraudulent transaction.

1. The mandates sent by the networks (such as VISA, MasterCard, and China UnionPay etc.,), which includes rules to avoid fraudulent transactions. The implementation of these rules is mandatory.
2. By analysing the historical transactions, the users can detect the frequent patterns of fraudulent transactions.

3. The industry norm.

Therefore, a user interface was created to enter the fraudulent transaction rules. The user has the capability to prioritize these entered rules according to the popularity of each rule. The motivation of prioritizing the rules by the user is that when the transaction is coming to the system, without entering to complex processes, the basic filtering is performed to check whether it is a possible fraudulent transaction or not.

3.9.2 Rule-based classification

The set of transactions which have been classified as non-fraudulent will be further processed. In this step, the user-entered rules are further improved.

The basic steps of rule-based classification algorithm are as follows.

1. Searching for rules that suit the most.
2. Remove the positive examples covered by this rule – Since by prioritizing the rules, the algorithm will search the records for the highest ranked rules initially. It will remove the records covered by a particular rule.
3. Repeat until the transaction has gone through all the rules.

To perform the rule-based classification, the RIPPER algorithm in the Sequential Covering technique is used. Due to its high performance compared with other algorithms and rapid classification ability of new instances, it has been selected.

For training of the model, these rules are inserted directly into the database. The rules that were defined in the training phase will also be used in the testing phase. Please refer section 4.2.1.1 for the entered.

3.9.3 RIPPER algorithm

To implement the Sequential Covering technique, the RIPPER algorithm [57] which is an acronym for “repeated incremental pruning to produce error reduction” has been used. As shown in Figure 7.1, the RIPPER sorts the classes in the ascending order based on the class size. For each of the user-entered rules, the algorithm grows the rules greedily, reconsiders the rule and two variants are produced. Using reduced-error pruning, instances covered by other classes are removed. If one of the two variants yield a better result, it will replace the rule with the new rule. These rules will be arranged according to ascending order of class size. The RIPPER algorithm is implemented using the JRip in Weka. The algorithm was run with 10-fold cross validation testing technique.

The rules obtained by JRip are listed below with the relevant description of each rule.

1.(trxn_amt > 80000) and (crd_acpt_id_code = 147) => group=1 (34.0 / 2.0)

Description - if the transaction amount is greater than 80000 and the card acceptor id = 147 (Group 1)

2. (MCC = 7995) and (crd_acpt_id_code = 414) => group=2 (10.0 / 2.0)

Description - if Merchant Category Code = 7995 and the card acceptor id = 414 (Group 2)

3. (MCC = 3280) and (crd_acpt_id_code = 070) and (added_time = 18:57) => group=3 (66.0 / 3.0)

Description - if Merchant Category Code = 3280 and the card acceptor id = 070 and transaction added time = 18:57 (Group 3)

4. (acqr_inst_id = 008412) and (crd_acpt_id_code = 012) => group=4 (42.0 / 2.0)

Description - if Acquirer Institute ID = 008412 and card acceptor id = 012 (Group 4)

5. (acqr_inst_id = 001660) and (MCC = 1558) => group=5 (4.0 / 0.0)

Description - if Acquirer Institute ID = 001660 and Merchant Category Code = 1558 (Group 5)

6. (term_id = F2203) and (crd_acpt_id_code = 015) => group=6 (5.0 / 0.0)

Description - if terminal id = F2203 and card acceptor id = 015 (Group 6)

An experiment was performed to compare the influence on the efficiency of the performance based on user-entered rule prioritization, the datasets were run on the same algorithm with and without the rule prioritization. The set of rules that were used for this task is specified in Section 4.2.1.1. The results obtained for the three datasets, mean processing time and standard deviation figures are given in. Table 3.2

As shown in the Table 3.2, when the rules entered by the users are prioritized, the processing becomes more efficient. The reason for this is, the users generally know the popular fraud patterns, such as transactions with very high values or transactions that are frequently occurring in a given period of time are more prone to be frauds. Therefore, to prevent further processing inside the FDM, the users can set the priority to these rules, in the descending order from most popular to least popular card fraud patterns.

Table 3.2: Rule prioritization - Influence on efficiency

		Rule-based classification with rule prioritization. (in seconds)	Rule-based classification without rule prioritization. (in seconds)
Dataset 1	**Mean**	1.060	1.129
	Std. deviation	0.089	0.093
Dataset 2	**Mean**	1.062	1.120
	Std. deviation	0.098	0.090
Dataset 3	**Mean**	1.063	1.122
	Std. deviation	0.092	0.087

Comparison of mean processing times, showing the effect of rule prioritization for the processing time of the algorithm is given in Figure 3.2. There is a clear reduction in mean processing times for all the three datasets and it could be concluded that rule

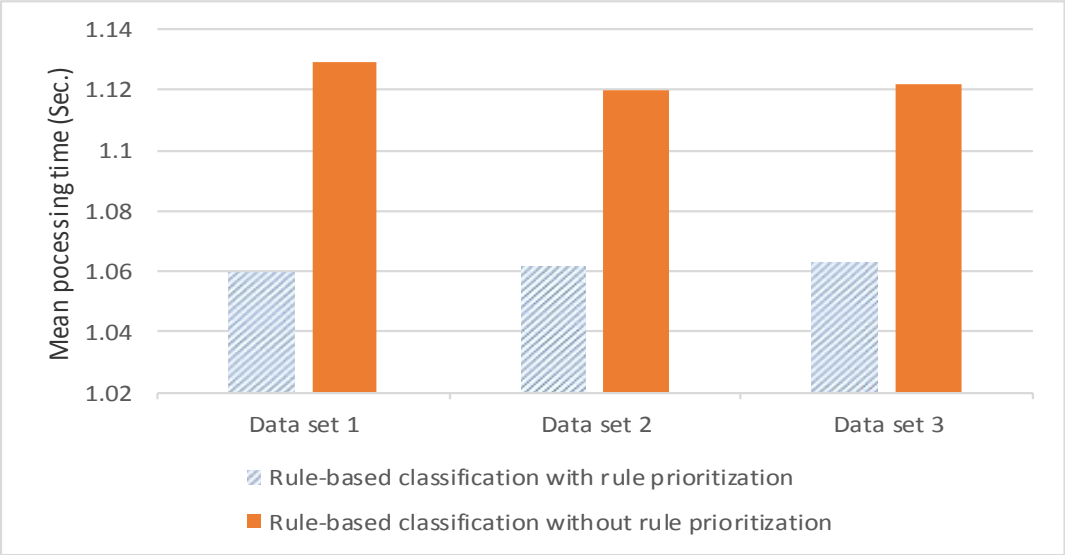

Figure 3.2: Effect of rule prioritization

The rule prioritization allows the users to prioritization. After the rule-based classification, the transactions that get classified as “fraudulent” will be blocked. The rest of the transaction data will be further processed by the model (See Figure 3.3) to verify whether they are suspicious or not

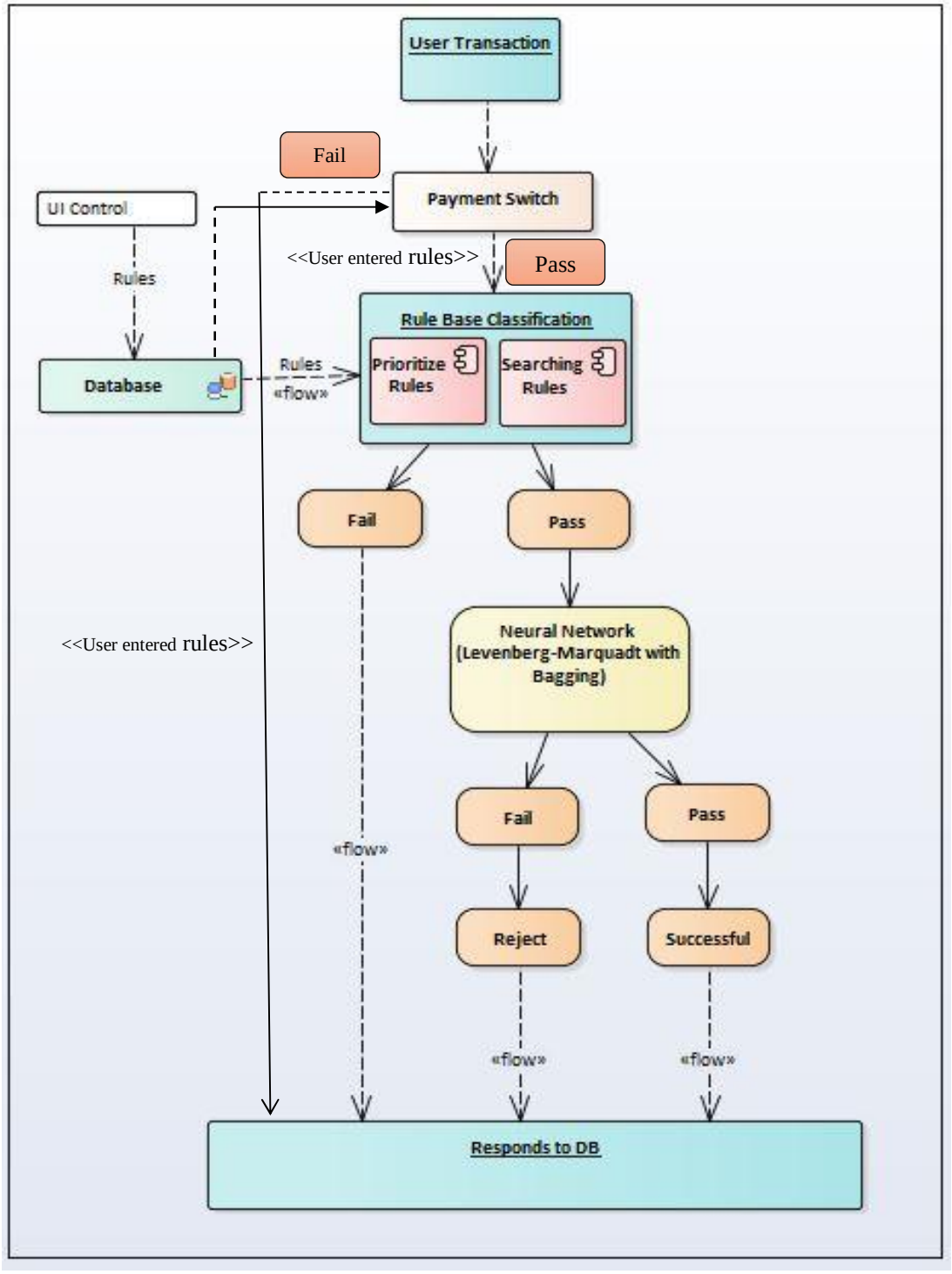

Figure 3.3: Architecture of the proposed ensemble model

3.9.4 Bagging with Levenberg-Marquardt neural networks algorithm

The popular ensembles methods that were in use are Bagging and Boosting. Bagging is always more accurate than an individual classifier and more resilient than Boosting [59]. Boosting is more sensitive to the noise which is partly to blame for its predicted error rate whereas Bagging produces a consolidated model which can give more accurate output [26]. Bagging minimizes the error from variance by averaging the bootstrap samples.
It is crucial for the model to give a fast output, the Bagging does this by training the classifier in different bootstrap and aggregating the outputs of those classifiers and give the final output by majority voting. Therefore, the Bagging method has more advantages in parallel computing than the other ensemble methods.

The Bagging algorithm is described in the Literature review in section 2.5.1.
The Bagging ensemble is used with the Levenberg-Marquardt neural networks algorithm due to its ability to provide the results with high accuracy as already discussed in section 3.8.

The tools used for the implementation to check the feasibility of the ensemble method is Matlab and Weka. The feasibility of analysis on Matlab is used with Weka GUI properties. Therefore, to transfer data back and forth the *matlab2weka* interface was used [19]. For the parallel processing of Weka in Matlab, the *runParallelWeka* library was used.
The Levenberg-Marquardt algorithm is trained using the *trainlm()* function in Matlab [19]. For bagging the *bag()* function in MatLab is used.
For the implementation of the neural network, a three layer network was used with one input layer, one hidden layer, and one output layer. The hidden layer is capable of approximate any function that contains a mapping from one finite space to another.

There is no strict formula for selecting the optimum number of hidden neurons. However, if *n* is the number of input neurons and *m* is the number of output neurons, the nodes in the hidden layer [37] (N_h) would be; N_h= sqrt (*n***m*)

Therefore, in this network, the number of inputs are 14 nodes (14 attributes) (refer 3.7) and the number of output parameters are 2 nodes (fraudulent or non-fraudulent), the number of hidden neurons are, sqrt (14*2) = 5 hidden nodes.

Another study [63] suggested that the number of hidden neurons are equal to the arithmetic mean, $N_h = (m+n)/2$. Which is equal to 8 neurons in the hidden layer. So, a combination of two methods suggests selecting between 5 – 8 hidden neurons for better results. It has been found that 7 hidden nodes provide the best solution.

3.10 Implementation

Prior to building the ensemble (FDM) model and to determine the accuracy, the data was subjected to training and testing of each of the base algorithm, Rule-based classification (Sequential Covering) and artificial neural network (the basic LM methodology), individually. The same rules are used for testing the FDM model, and the set of rules are given in the Section 4.2.1.1 Rules. Ten cross-validation folds were used. The results are analyzed and evaluated in the results and evaluation chapter.

4. RESULTS AND EVALUATION

4.1 Goal

The goal of this chapter is to test the trained FDM model, analyze and interpret the results to determine whether the following objectives of developing the model is achieved.

1. **Accuracy** –
   Whether this model can detect fraudulent transactions with a high accuracy.
2. **Efficiency –**
   Whether this model can detect fraudulent transactions in a timely manner.

4.2 Testing Strategy

The FDM model was tested on a computer with Intel(R) Core(TM) i7-6785 3.90GHz processor and 8.00GB of RAM.

There are three datasets containing a total of 5,700,000 recent payment card transactions from three financial institutions which include local and international transactions. The datasets contain card transactions spanning over a period of three years. Details of the test datasets are given in Table 4.1.

Table 4.1: Details of the test datasets

	Dataset 1	Dataset 2	Dataset 3
Number of records	2,000,000	2,300,000	1,400,000
Number of frauds	100,000	49,750	21,500

To determine the best results with the actual model, the transactions were pumped into a transaction switch using a transaction simulator, which then directs the transactions data to the fraud detection module as shown in Figure 4.1.

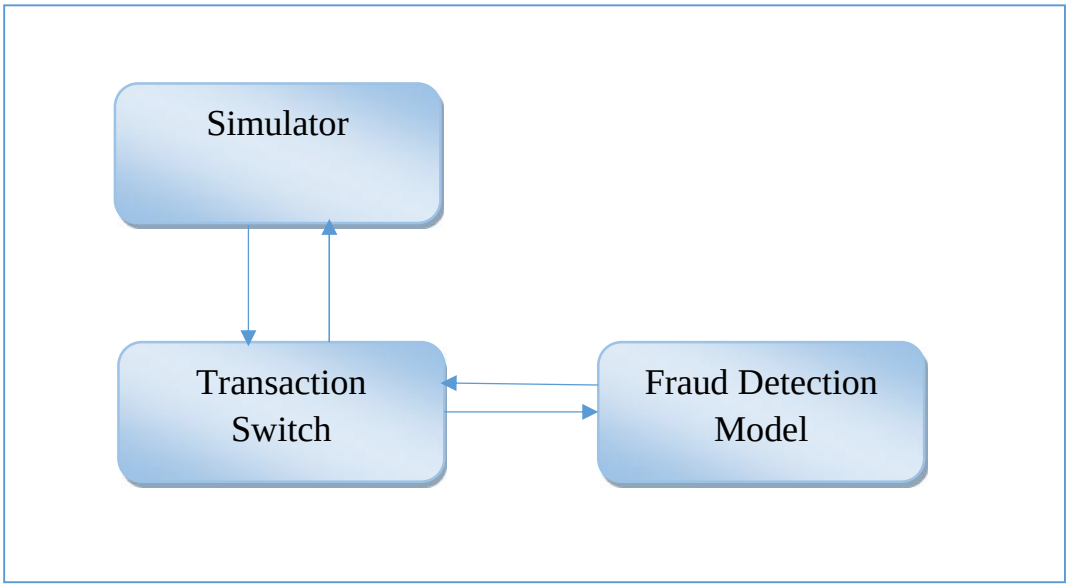

Figure 4.1: Test setup

The total time that a transaction spends inside this setup is approximately 10 seconds.

The specifications of the transaction switch have been the Linux RHEL 6.5 running on IBM Power S812LC server with 8-core, 32GB DRAM processor and 400GB of Hard disk.

4.2.1 Specified Rules

As discussed earlier, the rules should be inserted for the rule-based filtration. Figure 4.2 shows the interface that these rules were entered. It is a Java-based GUI that can be used to enter the rules by the user, which will get inserted into the database table directly. The classifiers will be extracting the rules from this database table.

4.2.1.1 Rules

Following are the rules that were inserted based on MasterCard Release 16.Q4 Mandate [36] guidance:

1. *Number of transactions per hour* – if the number of transactions is very high, there is a possibility that the transactions are fraudulent
2. *Last 10 transaction sum* – if the last 10 transaction sum is very high or very low, then those transactions can be suspicious
3. *Bill currency code* – if the bill currency code varies to about 3 types within a short time, then those can be suspicious transactions
4. *Bill Amount* – if the bill amount is very large or very small, such transactions are suspicious
5. *Acquirer Institute Country Code* – Suspicious country codes are specified (e.g.: Somalia)
6. Merchant Category Code – Suspicious merchant category codes are specified
7. Card Number – Suspicious card numbers, in which the transactions should be alerted can be specified
8. Velocity – if one transaction is done from Sri Lanka, and the next transaction is done from Germany within an hour, that indicates the location of the card has changed in an impossible velocity. Therefore, those transactions should be blocked.

The above criteria have been defined in Figure 4.2, the GUI interface for entering validation rules.

Element Id	Reference Id	Sign	Reference Value
101	LAST 10 TRXN SUM	>	1000000
102	BILL_CURR_CODE	=	HKD
121	VELOCITY	>	150
141	#TRXN / HOUR	>	5
161	BILL_AMT	=	75
181	ACQR_INST_CTRY_CODE	=	AFG

Figure 4.2: GUI interface for entering validation rules

4.3 Test results analysis

Prior to building the base-line for the model and to determine the accuracy, the data is subjected to training and testing of each of the algorithm individually. The above set of rules (section 4.2.1 Specified Rules) have been used to train Rule-based classification and neural network as well as the FDM model.

For testing the datasets Ten-fold cross validations testing technique was applied. The results are averaged to produce a single estimate.

4.3.1 Evaluation of the results of three test datasets

Using the FDM model, the results obtained performing tests with the three datasets are discussed here with a view in mind to evaluate its performances.

Performance metrics are used to evaluate the quality of classification of the FDM. Some of the popular metrics used for evaluation of binary classification are accuracy, recall, sensitivity, precision, F-measure, and ROC area. It is not desirable to rely on a single metric to decide a performance of an algorithm since no single metric can indicate all the desirable aspects.

Results obtained for the Dataset 1, 2, and 3 are shown in Table 4.2, Table 4.3, and Table 4.4 respectively after executing the new fraud detection model (FDM).

Table 4.2: Results of Dataset 1

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.924     0.235      0.884     0.924     0.904      0.939    c0
                 0.765     0.076      0.839     0.765     0.8        0.939    c1
Weighted Avg.    0.87      0.181      0.869     0.87      0.868      0.939
```

Table 4.3: Results of Dataset 2

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.894     0.088      0.952     0.894     0.922      0.958    c0
                 0.912     0.106      0.816     0.912     0.861      0.958    c1
Weighted Avg.    0.9       0.094      0.905     0.9       0.901      0.958
```

Table 4.4: Results of Dataset 3

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.909     0.235      0.882     0.909     0.896      0.896    c0
                 0.765     0.091      0.813     0.765     0.788      0.896    c1
Weighted Avg.    0.86      0.186      0.859     0.86      0.859      0.896
```

In this evaluation, several abbreviations have been used in important metrics such as TP (true positive). TN (true negative), FP (false positive), and FN (false negative). The data points related to them are given in the confusion matrix of binary classification shown in Table 4.5.

Table 4.5: Confusion matrix of binary classification

		Predicted data points	
		Positive	Negative
Expected data points	True	TP	TN
	False	FP	FN

True positive rate (TP rate) is the proportion of positive cases that were correctly identified with respect to all positive data points, as calculated by the equation; TP rate = TP / (TP + FN).

TP rate is also known as sensitivity, hit rate and recall. The high proportion of TP rate in all three tests indicates the high capability of the model in identifying data relevant to the criteria of the model.

False positive rate (FP rate) is the proportion of the number of data points identified positive that is actually negative with respect to all negative data points. FP rate is calculated by the formula, FP = FP / (FP + TN). It is the false alarm rate and the very low value in all three tests that indicates again the accuracy of identification of data correctly.

True negative rate (TN rate), which is popularly known as specificity indicates how good the test is at avoiding false alarm. The TN rate is calculated by the formula, TN = TN / (TN + FP). The rate is over 75% in all the cases prove that the high degree of accuracy in the classification.

Precision, which is also referred to as positive predictive rate is the proportion of data points retrieved that are truly of a class divided by the total data points classified as that class. So, it is calculated by Precision = TP / (TP + FP). In other words, precision indicates which frequency of the predictions is accurate. Hence it is a form of accuracy as well. So, high values of all the three test results indicate a high degree of relevancy in the selection.

Recall is the proportion of data points classified as a given class divided by the actual total in that class, so it is equivalent to TP rate. Recall indicates how complete the results or the probability that a relevant data is retrieved.

F-measure is a single measure of search effectiveness. It is a combined metric between precision and recall or in other words harmonic mean of precision and sensitivity which is calculated as, F-measure = 2 * Precision * Recall / (Precision + Recall). In data analysis, a classifier may achieve a higher precision and be highly accurate but at a cost of losing a significantly higher percentage of misclassified instances with a low recall rate. Therefore, recall together with precision provides more meaningful insight into the capability of the model in delivering expected results. However, in this analysis more intuitive than F-measure alone is, of course, reporting precision and recall separately. Both results have been considered above.

The reported F- measure ranging between 86% - 90% indicates how precise the classifier selecting the test instances correctly without missing a significant number of instances.

ROC (Receiver Operating Characteristics) area measure is the area under the curve (AUC) drawn TP rare (sensitivity) against FP rate (1- specificity) which quantifies the overall ability of the model to discriminate between correctly identified data points (true positive) from those with irrelevant (falls positive) data. If AUC is 1 then the prediction is perfect whilst if it is 0.5 then the prediction is random. Looking at the area under curve scores ranging between 90% - 96%, it can be concluded that the accuracy of the results is very high and there is no indication that they are predicted randomly.

Accuracy is a single metric for measuring the overall performance of the model. It is the proportion of all the predictions that are correctly classified given by the formula, Accuracy = (TP+TN) / (TP+TN+FP+FN). Sensitivity and specificity rates can be multiplied by their respective class size and weighted average to derive the same results. So, accuracy is a combined measure of sensitivity and specificity.

The weighted average of sensitivity and specificity rates of the three classifications ranging between 86% - 90% indicates the high accuracy of predictions of the model.

4.3.2 Comparison of classification accuracy of the model with base classifiers

For comparison of accuracy, the three datasets were also run on the Rule-based classification and the neural networks, the algorithms which were used to build the FDM model. As discussed earlier, accuracy is equal to the weighted average of sensitivity and specificity, which has been used for the comparison.

The screenshots of the results obtained for dataset one for Rule-based classification (Sequential Covering) and artificial neural network (the basic LM methodology) are shown in Table 4.6 and Table 4.7 respectively. The results obtained for the three test datasets with the FDM were already discussed and shown in Table 4.2, Table 4.3, and Table 4.4.

Table 4.6: Results of Rule-based classification with Dataset 1

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.758     0.441      0.769     0.758     0.763      0.724    c0
                 0.559     0.242      0.543     0.559     0.551      0.724    c1
Weighted Avg.    0.69      0.374      0.692     0.69      0.691      0.724
```

Table 4.7: Results of Neural network with Dataset 1

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.891     0.394      0.814     0.891     0.851      0.741    c0
                 0.606     0.109      0.741     0.606     0.667      0.741    c1
Weighted Avg.    0.794     0.297      0.789     0.794     0.788      0.741
```

The accuracy of the results that were obtained from each of the tests is shown in Table 4.8.The FDM model has 18%, 21%, and 16% higher accuracy compared with Rule base classification and, it has 8%, 9% and 8% higher accuracy rates compared with neural network for the Dataset 1, 2, and 3 respectively.

Table 4.8: Accuracy of the model and its base classifiers

	Rule-based classification (Sequential covering)	Neural network (basic LM)	Ensemble model
Dataset 1	0.69	0.79	0.87
Dataset 2	0.69	0.81	0.90
Dataset 3	0.68	0.78	0.86

Higher accuracy rates on the tested three datasets show the strength of the FDM model compared with its base classifiers, Rule base classifier and neural networks. The comparison is clearly visible from the graph in Figure 4.3.

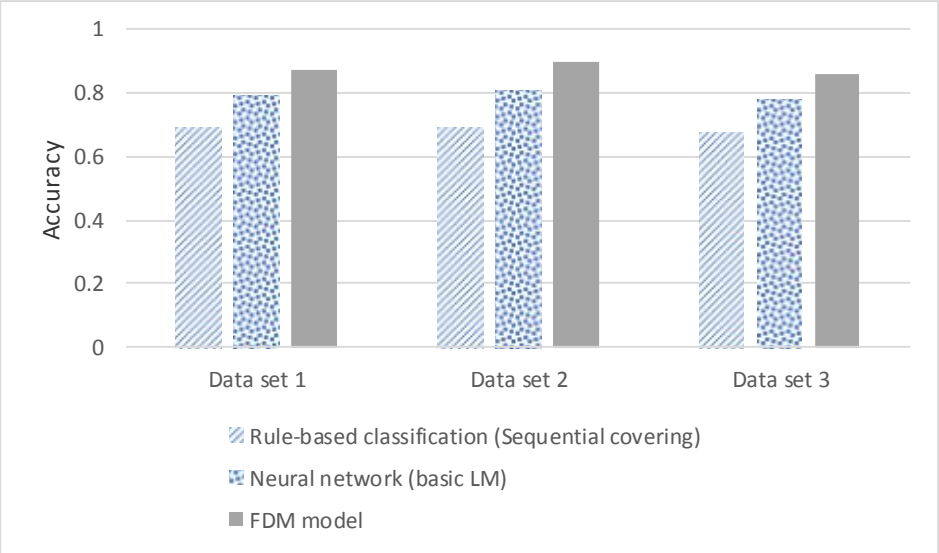

Figure 4.3: Accuracy of the model and its base classifiers

4.3.3 Review of accuracy of the model

The primary objective of developing the FDM model was to obtain results with a high accuracy. As explained overall accuracy is the weighted average of a classification's sensitivity and specificity. The accuracy obtained for the Dataset 1 was 86.9%. The accuracy of the Dataset 2 was 90.5% and for the Dataset 3, it was 85.9%. The values obtained for accuracy can be concluded as very impressive compared with the performance of the existing models available in the market, ranging between 70% - 75%.

It has been noticed that a few percentage of the instances are misclassified at the test. In the rule “velocity” (see section 4.2.1.1), there were approximately 2,300 transactions that were misclassified. This can be due to very high-speed change of location, the actual transaction may be legitimate, for example, the cardholder travelling from Sri Lanka to the Maldives. Though it is a legitimate transaction, it does not have a clear difference between the other illegitimate transactions.

For the rule “last 10 transaction sums” (see section 4.2.1.1), approximately 3,400 misclassified transactions were observed. The reason can be due to the series of ten

transactions, with one transaction for a very high amount, making the sum of the last ten transactions very high.

The other defined rules did not have a significant number of misclassifications, indicating a high degree of accuracy in classification.

Furthermore, selecting relevant attributes is very important to maintain the accuracy of classification. On the other hand, there will be an optimal subset of relevant attributes which contributes to improving the performance or the efficiency of the classification process. Each attribute is independent of other attributes and failure to identify one or several relevant attributes may seriously affect the quality of the output.

So, these rules are user-defined, and the user has the ability to choose them or change the priority of the rules in order to maximize the output of the classification process. On the other hand, the neural networks can learn to improve the results with time.

4.3.4 Comparison of efficiency of the model with base classifiers

The second goal of developing an FDM model was to have an efficient model. The efficiency of the FDM model has been determined using the time that the model used to process a single transaction. To achieve this, the same transactions were sent through the transaction switch (the specifications of the transaction switch is mentioned in Section 4.2). The time is derived using the response time to the transaction switch from the model for each of the transactions. The response time value is printed in the transaction switch log under <RSP_TIME> tag (Figure 7.2). Therefore, this tag is filtered out from the application log for each transaction and the mean and the standard deviation are calculated to compare the efficiency of each model.

The efficiency of the Rule-based algorithm and neural network are also derived while deriving the efficiency for the FDM model. Mean processing time for each algorithm with standard deviation is shown in Table 4.9: Efficiency of the

algorithms. When considered the distribution of average processing times (See Figure 4.4), Rule-based algorithm appears to be highly efficient with Ensemble (FDM) model being the slowest processor.

Table 4.9: Efficiency of the algorithms

		Rule-based classification (in seconds)	Neural network (basic LM) (in seconds)	FDM model (in seconds)
Dataset 1	**Mean**	1.06	1.37	2.51
	Std. deviation	0.08	0.01	0.18
Dataset 2	**Mean**	1.06	1.38	2.62
	Std. deviation	0.09	0.02	0.17
Dataset 3	**Mean**	1.06	1.36	2.42
	Std. deviation	0.09	0.01	0.16

The ensemble (FDM) model spends more time on processing due to each transaction having to pass through several algorithms inside the model. If it is possible to reduce the execution time of neural network, the efficiency of the model may increase considerably.

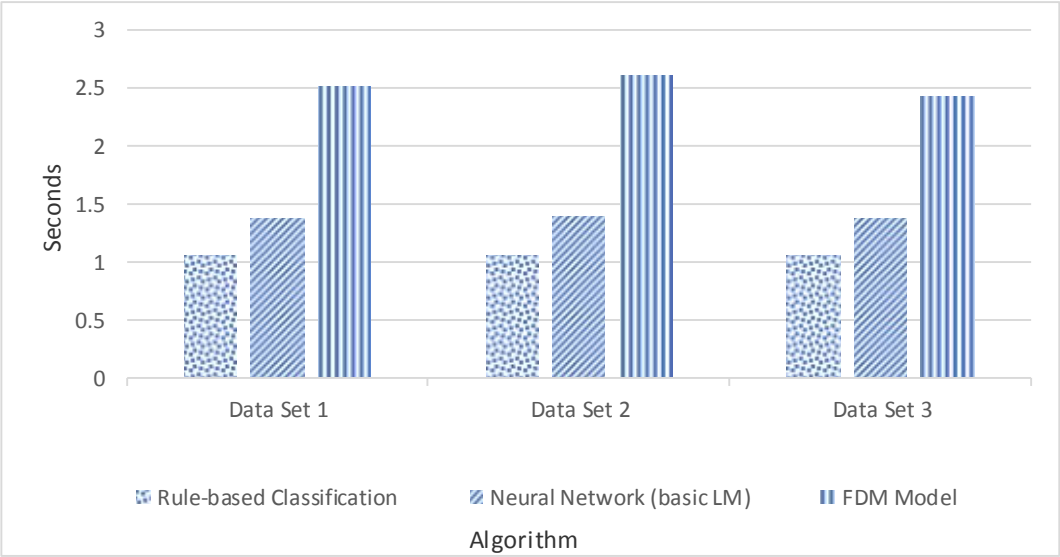

Figure 4.4: Mean processing times

During this experiment, the memory was kept constant. Higher execution time may be partly due to the resource limitations which could be overcome with alternative hardware solutions like parallel processing.

4.4 Performance comparison of the model with counterparts

Performance of the FDM model has been compared with the other popular algorithms used for payment card fraud detection. While the FDM model is an ensemble of the rule-based classifier and the neural network which is compared with single classifier counterparts, these different algorithms are run with the same set of test data.

Among the well-known decision tree techniques, C4.5 and CART are used for payment card fraud detection. Fraud detection models take the advantage of C4.5 algorithm`s capability of classifying new unseen instances or attribute values and its ability to handle missing attribute value. It can accept data with categorical or numerical values to construct a decision tree. The CART algorithm is also popular because of its ability to handle missing attributes and handling both categorical and

numeric values. By substituting surrogates for missing values, it generates robust and reliable predictive models even for very large databases with many variables and missing values. CART generate binary trees while C4.5 generate multi-branch trees. CART uses cost complexness pruning to remove the unreliable branches from the decision tree to improve the accuracy while C4.5 uses error based pruning strategy.

4.4.1 Accuracy

The results obtained for testing FDM model, Rule-based classifier and the neural network with the test Datasets were discussed in the previous section 4.3. The screenshots of the results that were obtained for the C4.5 and the CART algorithms with the Dataset 1 are shown in Table 4.10 and Table 4.11.

Table 4.10: Results of the C 4.5 algorithm with Dataset 1

```
=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0.636    0.059    0.955      0.636   0.764      0.551    0.731     0.827     c0
                0.941    0.364    0.571      0.941   0.711      0.551    0.731     0.495     c1
Weighted Avg.   0.740    0.162    0.824      0.740   0.746      0.551    0.731     0.714
```

Table 4.11: Results of the CART algorithm with Dataset 1

```
=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0.818    0.353    0.818      0.818   0.818      0.465    0.733     0.789     c0
                0.647    0.182    0.647      0.647   0.647      0.465    0.733     0.539     c1
Weighted Avg.   0.760    0.295    0.760      0.760   0.760      0.465    0.733     0.704
```

It has already been discussed that the ensemble FDM model is more accurate in its classification than its base classifiers, the Rule-based classifier and the neural network. The Rule-based classification ran with an average accuracy of 0.69, the neural network 0.79, the C4.5 algorithm 0.74 and the CART algorithm with 0.76. Being the highest among five models, the FDM model has achieved 0.87 of average accuracy. In Figure 4.5, a bar graph has been used for comparative display of accuracy of the results for each algorithm based on their weighted average of sensitivity and specificity.

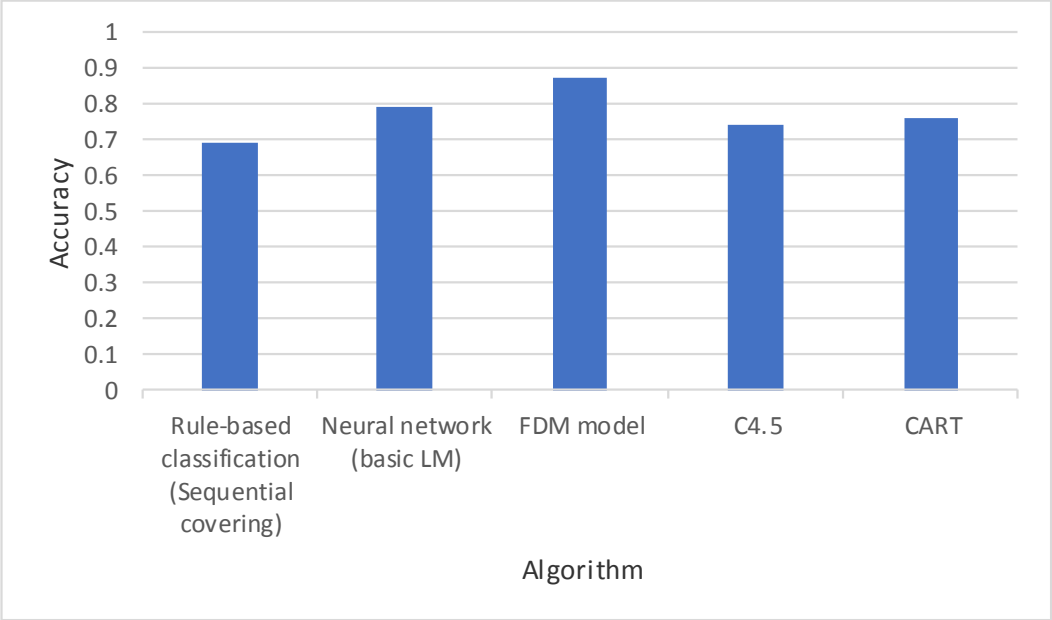

Figure 4.5: Comparative accuracy of FDM model with other algorithms

4.4.1.1 Testing for significance of accuracy

The above discussion led to prove that the FDM model detects frauds with better accuracy than the selected single classifiers. However, it is very important to determine whether the difference in accuracy is statistically significant or not.

The accuracy of the classifiers is defined as the proportion of samples correctly classified. So, the test of a hypothesis concerning a system of proportions could be applied to compare the accuracy of them for statistical significance.

If p_1 and p_2 are accuracy rates obtained from classifier 1 and 2 respectively for a test sample of size n, the standard error (SE) of the sampling distribution of difference between two accuracy rates is calculated as follows;

$SE = \sqrt{\frac{2p_0(1-p_0)}{n}}$ where p_0 is the pooled accuracy rate.

When classifiers are run on the same sample the pooled sample accuracy is given by,

$p_0 = (p_1 + p_2)/2$

So, the test statistic, z is given by,

$z = (p_1 - p_2)/SE$

The P-value is the probability of observing a sample statistic as extreme as the test statistic, which is compared with the significant level, and rejecting the null hypothesis when the P-value is less than the significant level.

Null hypothesis: $P_1 >= P_2$

Alternative hypothesis: $P_1 < P_2$

Note: p_1 denotes the accuracy rate of classifiers which is compared with FDM and p_2 denotes accuracy rate of FDM model.

Analysis of data and interpretation of results are given in Table 4.12.

Table 4.12: Testing for significance of accuracy

Classifiers comparing with FDM	Accuracy rate (p_1)	Difference in accuracy (p_1-0.87)	Z – Score for difference in accuracy	P -value	Decision at 0.01 significant level
NN	0.79	-0.08	-67.3	< 0.0001	Reject H_0
RB	0.69	-0.18	-137.4	< 0.0001	Reject H_0
CR5	0.74	-0.13	-103.8	< 0.0001	Reject H_0
CART	0.76	-0.11	-89.6	< 0.0001	Reject H_0

In all the cases, the P-value is less than 0.0001. Because P-value < 0.0001 ≤ α = 0.01, the null hypothesis is rejected. The results indicate that there is sufficient evidence at 0.01 significant level to conclude that FDM model performs better with respect to its accuracy in detecting fraudulent transactions compared with other four classifiers.

4.4.2 Efficiency

The mean processing times and standard deviations of the FDM model and the selected counterpart classifiers running on the Dataset 1 are given in Table 4.13. The transactions were sent through the payment switch (specifications of the transaction switch are mentioned in Section 4.2) to the fraud detection module which has been replaced with each of the following algorithms.

The average mean and standard deviation of the three datasets are shown and the values represent in seconds spent for a transaction.

Table 4.13: Comparison of efficiency of algorithms

	Rule-based classification (in seconds)	Neural network (basic LM) (in seconds)	FDM model (in seconds)	C4.5 (in seconds)	CART (in seconds)
Mean processing time	1.06	1.37	2.51	1.64	1.75
Std. deviation	0.09	0.01	0.18	0.06	0.24

The ensemble model shows the highest mean processing time of 2.51 seconds with 0.18 seconds of standard deviation indicating which is the least efficient compared with the other algorithms. The industry standard is that a transaction can spend 3 seconds inside a fraud detection model. Therefore, this efficiency raises no alarm. Nevertheless, there is a lot of room to improve the efficiency of the model. Also, it is worth to mention that the value of the efficiency of the ensemble model can be decreased by using parallel processing. To reduce the processing time as much as possible, the set of rules are prioritized and arranged in the descending order. The processing time would be high only for the transactions that do not get filtered from the Rule-based classification and will be passing through the rest of the ensemble model. Therefore, all the transactions may not always go through the neural network.

The comparison of mean processing times is graphically displayed in Figure 4.6.

The results of two sample t-tests for the comparison of above mean processing times indicate that at 95% confidence level the FDM model has a significantly higher mean processing time than the other four classifiers

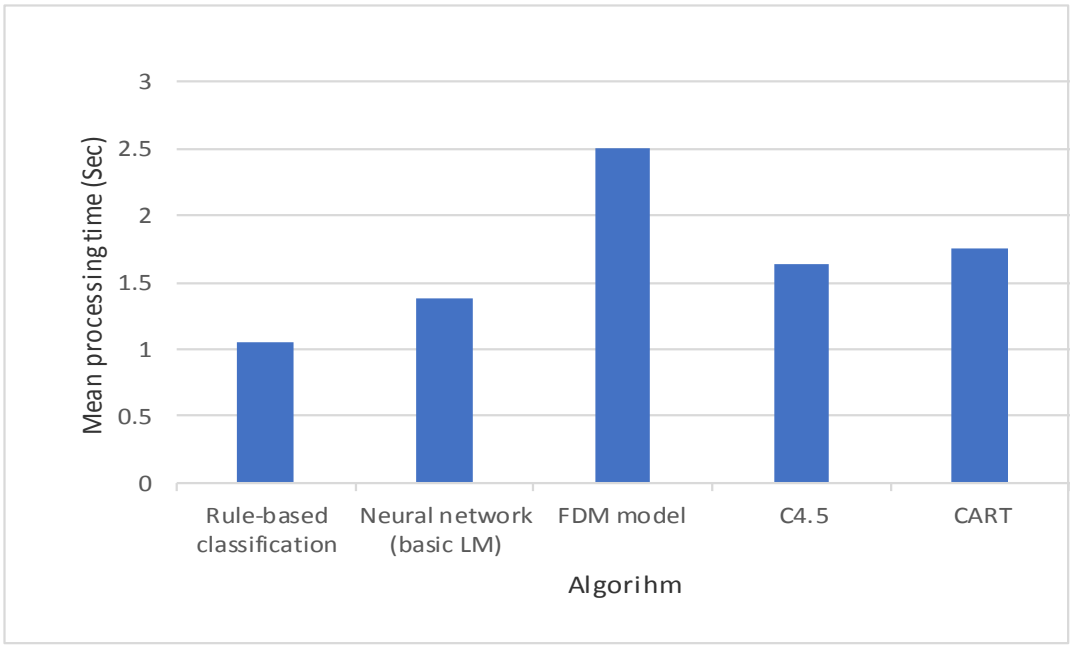

Figure 4.6: Mean processing times of FDM model and other algorithms

It seems that to gain high accuracy, the efficiency factor is compromised. In the real-world, financially the accuracy is very important compared with the processing time. Therefore, this is a favourable compromise.

However, the deficiency in processing time once the volume has increased could be overcome through load balancing, which distributes server loads across multiple resources. If there is a volume of transactions flowing into the system, a load balancer can be implemented from the server-level. Therefore, there will be two transaction switches, the main transaction switch, and the load balancing transaction switch together with their respective fraud detection module. When there is a high load of transactions coming into the system, the main server will route the excess transactions to the load balancing server for processing. The fraud detection model in each will perform its task with the incoming transactions and respond. Although financially expensive, this powerful technique provides a feasible solution to handle a large volume of transactions efficiently.

5. CONCLUSION AND FUTURE WORK

The ever-changing payment card frauds are explored and addressed in this study. After identifying the limitations of the industry, a new financial fraud detection model (FDM) was introduced. To be relevant to the current industry and to be efficient, a user-entered set of rules and enhanced set rules by rule-based classification are used as primary filters of the model. For further identification of the frauds, an ensemble of Levenberg-Marquardt neural network with bagging technique was used. The ensemble model works with a mean accuracy rate of 87.5%, while the existing models in the industry achieve around 70% of accuracy rate. The accuracy of the model could be further improved if the hidden pattern of frauds is identified. The efficiency of this ensemble model can be greatly improved by further optimizing the algorithms and executing in alternative processing environment from its current mean processing time which is 2.51 seconds per transaction.

An API layer can be built based on the FDM to implement cardholder alerts (e.g. SMS or E-mail), for transaction monitoring by financial institution personnel and to block / alert certain types of transactions after cardholder confirmation.

Also, the model should be modified to use in very large financial institutions with higher transaction processing capabilities, such as 200 transactions per second, which is a very large volume to be analyzed in a matter of seconds. Improvement may also be necessary to effectively handle classification problems with variable misclassification costs.

Furthermore, the model could be improved to have the capability of detecting fraudulent transactions in other financial sector institutions like insurance and tax.

6. REFERENCES

1. ‘About EMV’, *EMVCo*. [Online]. Available: http://www.emvco.com/about_emv.aspx. [Accessed: 17-Mar-2015].

2. A.Darwiche, ‘Bayesian Networks’, *Communications of the ACM*, vol. 53, no. 12, pp. 80–90, Jan. 2010.

3. A. Jain, M. Murty, and P. Flynn, ‘Data Clustering: A Review’, *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–322, Sep. 1999.

4. A. Marathe A, HA. Shawky,“ Categorizing mutual funds using clusters”, Advances in Quantitative Analysis of Finance and Accounting, vol. 7, 1999, pp.199–211

5. Brause, R., Langsdorf, T. and Hepp, M., Neural data mining for credit card fraud detection. In Tools with Artificial Intelligence, 1999. Proceedings. 11th IEEE International Conference on (pp. 103-106). IEEE., 1999.

6. ‘Card and Mobile Payment Industry Statistics | The Nilson Report Archive of Charts & Graphs’, *The Nilson Report*, Nov-2013. [Online]. Available: http://www.nilsonreport.com/publication_chart_and_graphs_archive.php?1=1. [Accessed: 17-Mar-2015].

7. C. Aggarwal and P. Yu, ‘Outlier detection for high dimensional data’, *ACM SIGMOD Record*, vol. 30, no. 2, pp. 37–46, Jan. 2001.

8. C.Lu, Y. Kou, and S. Sinvongwattana, ‘Survey of Fraud Detection Techniques’, in *2004 IEEE International Conference on Networking, Sensing & Control*, Taipei, Taiwan, 2004.

9. D. H. Fisher, "Knowledge acquisition via incremental conceptual clustering," Machine Learning, vol. 2, no. 2, pp. 139–172, Sep. 1987.

10. D. Sa´nchez et al., “Association rules applied to credit card fraud detection”, Expert Systems with Applications. vol.,36, no.2, pp. 3630–3640, March.2009. doi: 10.1016/j.eswa.2008.02.01

11. E.Aleskerov et al., "Cardwatch: A neural network based database mining system for credit card fraud detection", in *Proc. 1997 IEEE/IAFE Computational Intelligence for Financial Engineering (CIFEr), March, pp.220-226*

12. E.Duman and M.Oszelik, “Detecting credit card fraud by genetic algorithm and scatter search”, Expert Systems with Applications: An International Journal., vol.,38, no.10, pp. 13057-13063, September.2011. doi: 10.1016/j.eswa.2011.04.110

13. E.Duman et al., "A Novel and Successful Credit Card Fraud Detection System Implemented in a Turkish Bank", in *13th International Conf. Data Mining Workshops (ICDMW), 2013 IEEE*, pp. 162 – 171

14. E. Gately, *Neural networks for financial forecasting*, 1st ed. New York: New York : Wiley, c1996., 1995.

15. E.Kirkos et al., “Data Mining techniques for the detection of fraudulent financial statements” Expert Systems with Applications. vol.,32, no.4, pp. 995–1003, May. 2007. doi: 10.1016/j.eswa.2006.02.016

16. F. Bonchi, F. Giannotti, G. Mainetto, and D. Pedreschi, ‘A classification-based methodology for planning audit strategies in fraud detection’, *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’99*, pp. 175–184, Jan. 1999.

17. Fischthal, S. Neural network/conceptual clustering fraud detection architecture. US 5822741. 1998

18. *Fraud Prevention Issuer’s Best Practice Guide*, 03/11 ed. FIS Card Services, Inc., 2011.

19. H. Demuth and M. Beale, Neural Network Toolbox For Use with MATLAB®, Version 4 ed. The MathWorks, Inc, 2004.

20. H. Mizes, *Card Fraud*. Xerox Corporation. 2013.

21. Hilas, C. and Mastorocostas.P, An application of supervised and unsupervised learning approaches to telecommunications fraud detection. Knowledge-Based Systems, 21(7), pp.721-726, 2008

22. H. Wang, W. Fan, P. Yu, and J. Han, ‘Mining Concept-Drifting Data Streams Using Ensemble Classifiers’, in *Ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, Washington, DC, USA, 2003.

23. J. Schafer, "Package ‘norm’ title analysis of multivariate normal datasets with missing values," Feb. 20, 2015. [Online]. Available: https://cran.r-project.org/web/packages/norm/norm.pdf. Accessed: Apr. 1, 2016.

24. K. Yamanishi and J. Takeuchi, ‘Discovering outlier filtering rules from unlabeled data’, *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’01*, pp. 389–394, Jan. 2001.

25. K. Geras, "Prediction Markets for Machine Learning," 2011.

26. K.-H. Chang, "Complementarity In Data Mining," University of California, Los Angeles, 2015.

27. K. Yamanishi, J. Takeuchi, G. Williams, and P. Milne, ‘On-Line Unsupervised Outlier Detection Using Finite Mixtures with Discounting Learning Algorithms’, *Data Mining and Knowledge Discovery*, vol. 8, no. 3, pp. 275–300, Jan. 2004.

28. L. Breiman, "Bagging Predictors," Machine Learning, pp. 123–140, 1996.

29. L. Breiman, "Bias, variance, and arcing classifiers," UC-Berkeley, Berkeley, CA, 1996.

30. L. Breiman, J. Friedman, and C. J. Stone, Classification and regression trees. New York, NY: Chapman and Hall/CRC, 1984.

31. Lin, J.W., Hwang, M.I. and Becker, J.D., A fuzzy neural network for assessing the risk of fraudulent financial reporting. Managerial Auditing Journal, 18(8), pp.657-665. 2003.

32. Lourakis, M.I. A brief description of the Levenberg-Marquardt algorithm implemented by Levmar. Foundation of Research and Technology,4(1)., 2005.

33. MasterCard, "IPM Clearing Formats," 2015.

34. M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander, ‘LOF: identifying density-based local outliers’, *ACM SIGMOD Record*, vol. 29, no. 2, pp. 93–104, Jan. 2000.

35. M. Srinivas and L. M. Patnaik, ‘Genetic algorithms: a survey’, *Computer*, vol. 27, no. 6, pp. 17–26, Jun. 1994.

36. MasterCard Global Release 16. Q4 Document Dual Message and Single Message Systems. (2016) (1st ed.).

37. Masters, T. Practical neural network recipes in C++ (1st ed.). San Diego [u.a.]: Acad. Press, 1999.

38. Sci2s.ugr.es. (2017). Noisy Data in Data Mining | Soft Computing and Intelligent Information Systems. [online] Available at http://sci2s.ugr.es/noisydata#Introduction to Noise in Data Mining [Accessed 11 Feb. 2017].

39. Patent US5822741 - neural network/conceptual clustering fraud detection architecture, by S. Fischthal and L. M. Corporation. (1996, Feb. 5). [Online]. Available: http://www.google.com/patents/US5822741.

40. R.Bolton and D.Hand, “Statistical Fraud Detection: A Review” Statistical Science., vol.17, no.3, pp.235-255, January 2003. doi: 10.1214/ss/1042727940

41. Rdocumentation.org. (2017). da.norm function | R Documentation. [online] Available at: https://www.rdocumentation.org/packages/norm/versions/1.0-9.5/topics/da.norm [Accessed 22 January 2017].

42. S. Bay and M. Schwabacher, ‘Mining distance-based outliers in near linear time with randomization and a simple pruning rule’, in *Ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, Washington, DC, USA, 2003.

43. S.B.E. Raj and A.A.Portia, "Analysis on credit card fraud detection methods", *in Proc. 2011 International Conference on Communication and Electrical Technology (ICCCET)*, pp. 152 – 156

44. S.Ghosh and D.Reilly, “Credit Card Fraud Detection with a Neural-Network”, in Proc. of 27th Hawaii International Conference on Systems Science, 1994, pp.621-630.

45. Shen, A., Tong, R. and Deng, Y. Application of Classification Models on Credit Card Fraud Detection. In: Service Systems and Service Management, 2007 International Conference on. IEEE., 2007.

46. S.Maes et al., “Credit Card Fraud Detection Using Bayesian and Neural Networks,” in *Proc. 1st international naiso congress on neuro-fuzzy technologies*, January 2002.

47. Takeuchi and Yamanishi, ‘A unifying framework for detecting outliers and change points from time series’, *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 4, pp. 482–492, Jan. 2006.

48. T.Guo and G.Li, "Neural data mining for credit card fraud detection", in *2008 International Conf. Machine Learning and Cybernetics, July, pp.* 3630 – 3634.

49. T.Fawcett and F.Provost, “Combining Data Mining and Machine Learning for Effective Fraud Detection”, in Proc. *Second International Conference on Knowledge Discovery* and Data *Mining, 1996, pp.8-13. T.Guo and G.Li, "Neural data mining for credit card fraud detection", in 2008 International Conf. Machine Learning and Cybernetics, July, pp. 3630 – 3634.*

50. T. Fawcett and F. Provost, ‘Activity monitoring’, *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’99*, pp. 53–62, Jan. 1999.

51. T. Nguyen, J. Schiefer, and A. Tjoa, ‘Sense & response service architecture (SARESA): an approach towards a real-time business intelligence solution and its use for a fraud detection application’, *Proceedings of the 8th ACM international workshop on Data warehousing and OLAP*, pp. 77 – 86, 2005.

52. ‘Payment cards / Financial crime / Crime areas / Internet / Home’. [Online]. Available: http://www.interpol.int/Crime-areas/Financial-crime/Payment-cards. [Accessed: 20-Mar-2015].

53. P.K.Chan et al., “Distributed data mining in credit card fraud detection”, IEEE Intelligent Systems and their Applications., vol.14, pp. 67 – 74, Nov/Dec. 1999. doi: 10.1109/5254.809570

54. v1.0-9.5, n. (2017). norm package | R Documentation. [online] Rdocumentation.org. Available at: https://www.rdocumentation.org/packages/norm/versions/1.0-9.5 [Accessed 22 January 2017].

55. Wheeler, R. and Aitken, S. Multiple algorithms for fraud detection. Knowledge-Based Systems, 13(2), pp.93-99., 2000.

56. Whitley, D., genetic algorithm tutorial. Statistics and computing, 4(2), pp.65-85., 1994.

57. Witten, I., & Frank, E.). Data Mining Practical Machine Learning Tools and Techniques (2nd ed., pp. 403-468). Morgan Kaufmann publications., 2005.

58. Y. Amit and D. Geman, "Shape Quantization and recognition with Randomized trees," Neural Computation, vol. 9, no. 7, pp. 1545–1588, Oct. 1997.

59. Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of online learning and an application to boosting," Journal of Computer and System Sciences, vol. 55, no. 1, pp. 119–139, Aug. 1997.

60. Yoshida, K., Adachi, F., Washio, T., Motoda, H., Homma, T., Nakashima, A., Fujikawa, H. & Yamazaki, K. (2004). Density Based Spam Detector. Proc. of SIGKDD04, 486-493.

61. Zareapoor, M., Seeja.K.R, S. and Afshar Alam, M. (2012). Analysis on Credit Card Fraud Detection Techniques: Based on Certain Design Criteria. International Journal of Computer Applications, 52(3), pp.35-42.

62. Z. Zheng, "Boosting and Bagging of Neural Networks with Applications to Financial Time Series," 2006.

63. WITTEN, I.H.; FRANK, E. Data mining: Practical machine learning tools and techniques, Morgan Kaufmann, San Francisco, CA, 560 pp., 2005

7. APPENDICES

Appendix A – Steps of implementation of RIPPER algorithm

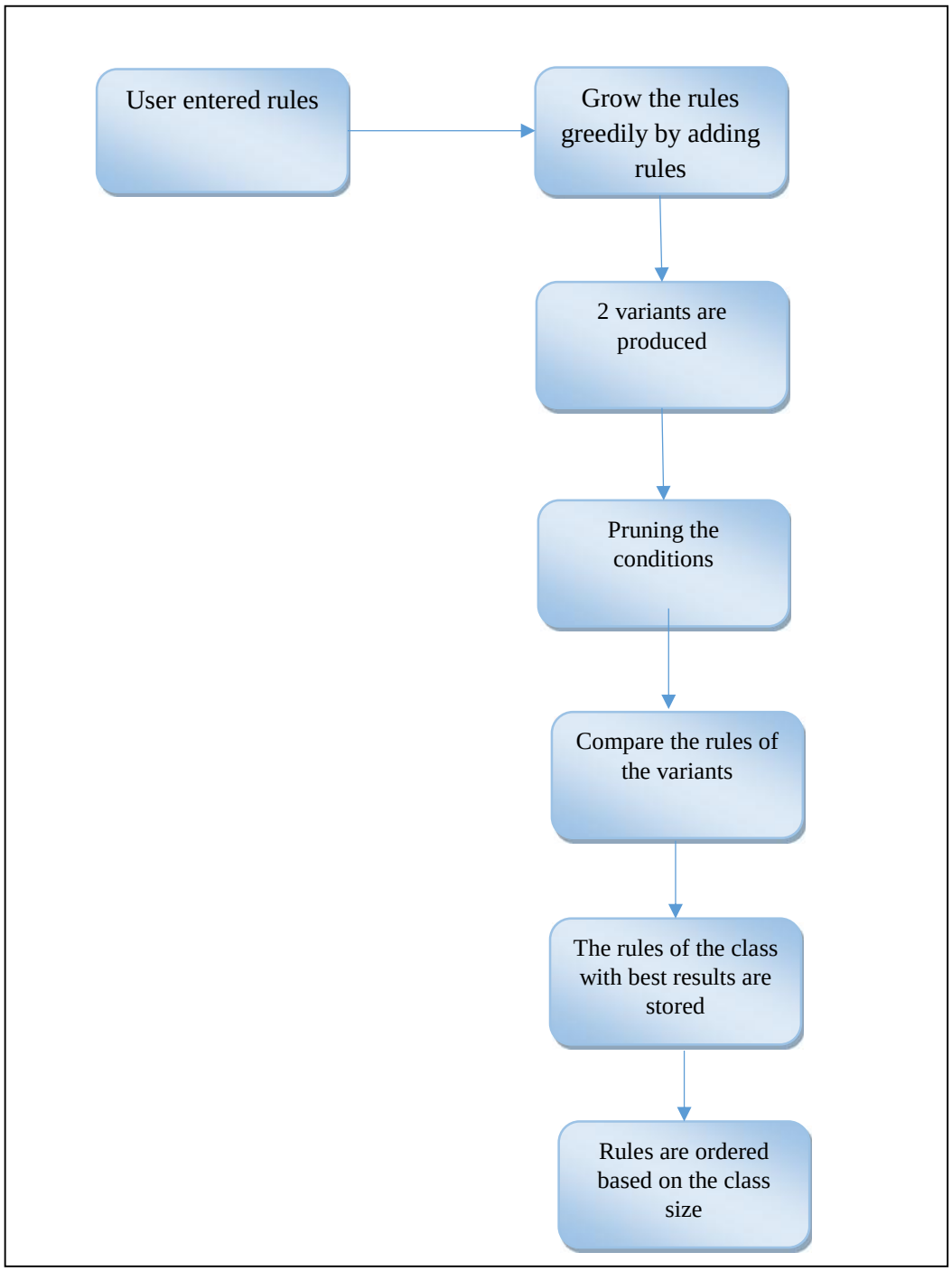

Figure 7.1: Implementation of RIPPER algorithm

Appendix B – Request and response messages of FDM

Following are the request and response messages that the transaction switch is sending to the ensemble model.

The Figure 7.2 shows the message that the transaction switch is sending to the fraud detection module.

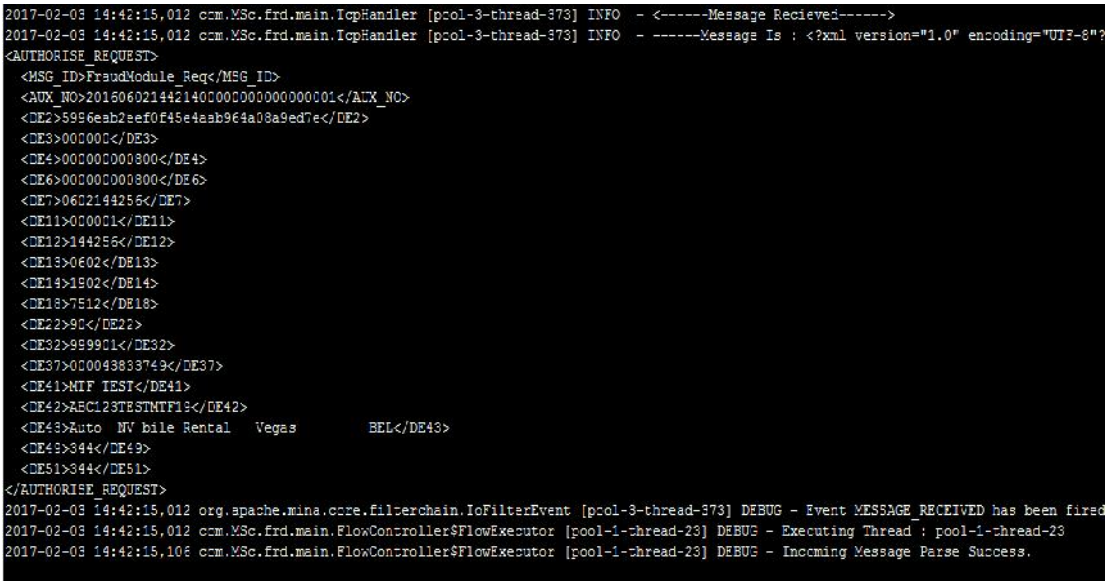

Figure 7.2: Request message from Switch

“DE” means data element as they are the attributes of the transaction. “DE2” represents the encrypted card number.

The Figure 7.3 shows the response message the fraud detection module has sent to the switch.

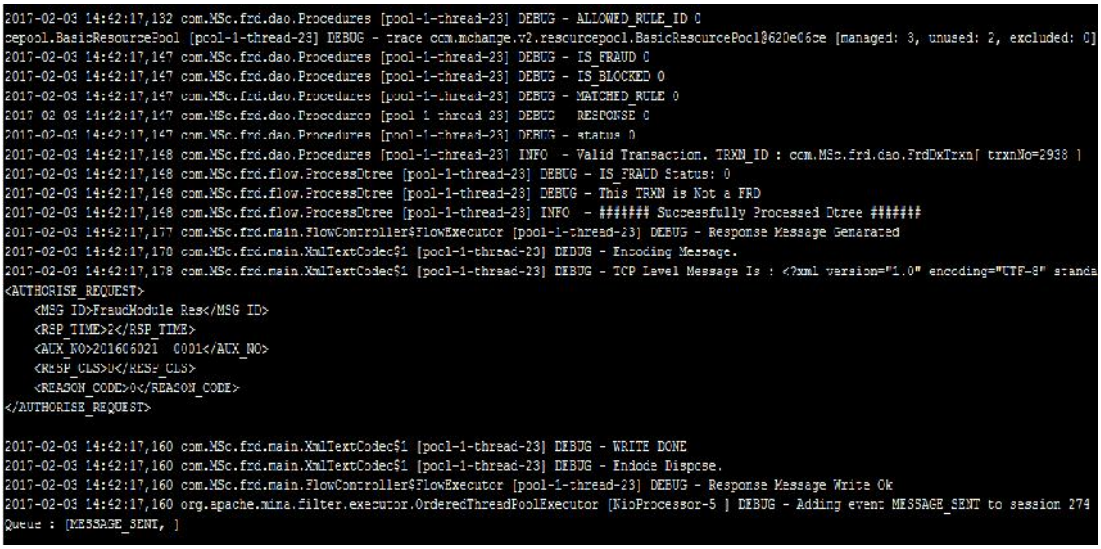

Figure 7.3: Response message to Switch

Fields of the log are as follows:

ALLOWED_RULE_ID – if the transaction was captured to a particular rule, this parameter shows to which rule id, the transaction has been captured.

IS_FRAUD – “1” if the transaction is a fraud, “0” if the transaction is legitimate.

IS_BLOCKED – “1” if the transaction is blocked, “0” if the transaction did not get blocked and passed through.

MATCHED_RULE - “1” if the transaction got matched to a rule, “0” if the transaction did not get matched to a rule.

RESPONSE – “1” if the transaction is invalid, “0” if the transaction is valid.

RSP_TIME – the time taken from the fraud detection module to respond to switch in seconds.

REASON_CODE – the message reason code, “0” if valid, if invalid this field will show the rule id in the fraud detection module.