

AN ABSTRACT OF A THESIS

AN EFFICIENT TECHNIQUE FOR MINING BAD CREDIT ACCOUNTS FROM BOTH OLAP AND OLTP

Sheikh Rabiul Islam

Master of Science in Computer Science

Credit card companies classify accounts as a good or bad based on historical data where a bad account may default on payments in the near future. If an account is classified as a bad account, then further action can be taken to investigate the actual nature of the account and take preventive actions. In addition, marking an account as "good" when it is actually bad, could lead to loss of revenue - and marking an account as "bad" when it is actually good, could lead to loss of business. However, detecting bad credit card accounts in real time from Online Transaction Processing (OLTP) data is challenging due to the volume of data needed to be processed to compute the risk factor. We propose an approach which precomputes and maintains the risk probability of an account based on historical transactions data from offline data or data from a data warehouse. Furthermore, using the most recent OLTP transactional data, risk probability is calculated for the latest transaction and combined with the previously computed risk probability from the data warehouse. If accumulated risk probability crosses a predefined threshold, then the account is treated as a bad account and is flagged for manual verification. In addition, our approach is efficient in terms of computation time and resources requirement because no transaction is processed more than once for the risk factor calculation. Another factor that makes our approach efficient is the early detection of bad accounts or fraud attempts as soon as the transaction takes place, which leads to a decrease in lost revenue.

**AN EFFICIENT TECHNIQUE FOR MINING BAD CREDIT ACCOUNTS FROM BOTH
OLAP AND OLTP**

A Thesis

Presented to

the Faculty of the College of Graduate Studies

Tennessee Technological University

by

Sheikh Rabiul Islam

In Partial Fulfillment

of the Requirements for the Degree

MASTER OF SCIENCE

Computer Science

May 2018

CERTIFICATE OF APPROVAL OF THESIS

**AN EFFICIENT TECHNIQUE FOR MINING BAD CREDIT ACCOUNTS FROM BOTH
OLAP AND OLTP**

by

Sheikh Rabiul Islam

Graduate Advisory Committee:

Sheikh Ghafoor, Co-chairperson

Date

William Eberle, Co-chairperson

Date

Doug Talbert

Date

Approved for the Faculty:

Mark Stephens
Dean
College of Graduate Studies

Date

DEDICATION

To my parents and wife.

ACKNOWLEDGMENTS

I would like to first thank both of my advisors, Dr. Sheikh Ghafoor and Dr. William Eberle, for their persistence and guidance all the way throughout this work. They always helped me with guidance whenever I ran into trouble and pushed me to gain the necessary technical and research skills. This research work would not have been possible without their unending supply of patience and stellar guidance.

I would also like to thank my committee member Dr. Doug Talbert for his support, encouragement, and feedback. His passionate participation and input encouraged me a lot to complete this work.

I would also like to thank the professors of the Computer Science Department I've had over the years. I specifically thank Dr. Ambareen Siraj, Dr. Gerald Gannod and Dr. Michael Rogers for providing me with unfailing support and continuous encouragement throughout my years of study and throughout this work. I would also like to thank Dr. Robert Qiu from the Electrical and Computer Engineering Department for his support and feedback.

I am grateful to my student colleagues and lab mates for their support and accompany. I am also grateful to our amazing university staff, Megan Cooper, Sydney Gebka, and Valerie Nash, for their support and assistance throughout my time at Tennessee Tech University.

Finally, I must express my gratitude to my parents and wife for their unfailing love and support. This accomplishment would not be possible without any of you. Thank you.

TABLE OF CONTENTS

	Page
LIST OF TABLES	vii
LIST OF FIGURES	viii
Chapter	
1. INTRODUCTION.....	1
1.1 Types of Fraud	2
1.2 Challenges in Financial Fraud Detection	5
1.3 Limitations	6
1.4 Machine Learning is a good fit for fraud mining	6
1.5 Some of the Applications of Machine Learning.....	7
1.6 Objective and Research Questions	8
1.7 Contributions.....	9
1.8 Organization.....	10
2. BACKGROUND AND LITERATURE REVIEW	11
2.1 Machine Learning Algorithms	11
2.1.1 Decision Tree Classification:.....	11
2.1.2 Decision Tree Regression.....	11
2.1.3 Linear Regression.....	12
2.1.4 Logistic Regression	13
2.1.5 Support Vector Machine.....	13
2.1.6 Fuzzy logic based system	14
2.1.7 Naïve Bayes.....	14
2.1.8 Random Forest	14
2.1.9 Extremely Random Trees	14
2.1.10 Artificial Neural Network.....	15
2.1.11 Deep Learning	16
2.1.12 K-means clustering.....	16
2.1.13 Principal Component Analysis	16
2.1.14 Hidden Markov Models.....	16
2.1.15 Genetic Algorithm (GA).....	17
2.1.16 Meta-Learning	18
2.1.17 Comparison of ML algorithms	18
2.2 Fraud	21

	Page
2.2.1 Financial Statement Fraud	21
2.2.2 Credit Card Fraud	23
2.2.3 Bankruptcy/Default on Payment Fraud.....	24
2.2.4 Other Fraud	26
2.2.5 Research by fraud type	29
2.2.6 Research by datasets	29
2.2.7 Research by Algorithms.....	30
2.2.8 List of attributes found in different dataset and research	32
2.2.9 Performance Evaluation Metrics.....	34
2.3 Summary	37
3. RESEARCH METHODOLOGY AND DESIGN	38
4. EXPERIMENT SETUP	41
4.1 Data	41
4.2 Technology Used.....	46
4.3 Experiments.....	47
4.3.1 Standard Transaction Testing	48
4.3.2 Customer Specific Testing.....	51
5. RESULT AND ANALYSIS.....	56
5.1 Result of experiment set I using Dataset I and II.....	56
5.2 Results for experiment set II using Dataset III	62
6. CONCLUSION AND FUTURE WORK	67
6.1 Future Work	67
6.1.1 Possible Improvements:	67
6.1.2 Future Research	68
REFERENCES	69
VITA	74

LIST OF TABLES

Table	Page
Table 1. Advantage and disadvantage of different algorithms	19
Table 2. Research by fraud type	29
Table 3. Research by datasets.....	30
Table 4. Research by algorithms	31
Table 5. List of attributes	32
Table 6. Evaluation metrics.....	34
Table 7. Dataset I	41
Table 8. Standard Rules.....	49
Table 9. Sample OLTP data	49
Table 10. Relevancy Mapping.....	50
Table 11. Customer Specific Rules	51
Table 12. Training Time (Dataset I).....	56
Table 13. Computation Time (Dataset I).....	57
Table 14. Accuracy, Precision, Recall, and F-score	58
Table 15. Suspicious transaction infusion	61
Table 16. Batch wise performance metrics on Dataset III.....	64
Table 17. Comparison of the result (Direct approach vs Proposed approach)	64

LIST OF FIGURES

Figure	Page
Figure 1. Categories of financial fraud	3
Figure 2: Most common credit card frauds.....	3
Figure 3: Linear Regression [41][42]	12
Figure 4: Support Vector Machine [42].....	13
Figure 5: Artificial Neural Network	15
Figure 6. A high-level diagram of the proposed approach	38
Figure 7. Flowchart of the proposed approach	39
Figure 8. Dataset II	42
Figure 9. Dataset III (Taiwan dataset) part I.....	43
Figure 10. Dataset III (Taiwan dataset) part II	43
Figure 11. OLAP dataset created from dataset III	44
Figure 12. OLTP dataset created from dataset III.....	44
Figure 13. Spain Dataset.....	45
Figure 14. Range determination using equal frequency binning.	46
Figure 15. Feature Selection	47
Figure 16. Probability Distribution.....	52
Figure 17. Training Time (Dataset I).....	57
Figure 18. Computation Time (Dataset I).....	58
Figure 19. Accuracy, Precision, Recall, and Fscore (Dataset I)	59
Figure 20. Flagging accounts for verification.....	60
Figure 21. Bad accounts visualization	60
Figure 22. Performance by algorithms on the whole dataset (before decomposition).....	63
Figure 23. Batch wise performance metrics on Dataset III.....	64
Figure 24. Performance comparison of different approaches	65
Figure 25. Batch size vs computation time	66

CHAPTER 1

INTRODUCTION

Credit cards are usually issued by a bank, business or other financial institution that allow the holder to purchase goods and services on credit. A person can have multiple credit cards from different companies. Companies who provide credit scores suggest cardholders use multiple credit cards in order to increase their credit score. A credit score is a three-digit number between 300 and 850 that indicate the creditworthiness of a person. The credit score is used by lenders to determine someone's credit worthiness for various lending purposes.

A credit score can affect whether or not someone is approved for credit as well as what interest rate they will be charged [48]. Recklessly using multiple credits card is one of the reasons that someone is unable to pay their credit card bill on time, which can eventually turn into long-term debt for the cardholder. Other reasons for being unable to pay their bill include job loss, health issues, or an inability to work, which can eventually result in "bankruptcy ". In any case, this becomes an issue for both the credit card companies and the customer.

To address this problem, besides carefully evaluating the creditworthiness of credit card applicants at the very beginning, the credit card issuer needs to identify potential bad accounts that are at the risk of going into bankruptcy over the life of their credit. From the creditor's side, the earlier the bad accounts are identified, the lower the losses [13]. A system that can identify these risky accounts in advance would help credit card companies to take preventive actions. They could also potentially communicate information to the account holder and provide suggestions for avoiding bankruptcy.

Online Analytical Processing (OLAP) systems typically use archived historical data from a data warehouse to gather business intelligence for decision-making. On the other hand, Online Transaction Processing (OLTP) systems, only analyze records within a short window of recent activities - enough to successfully meet the requirement of current transactions [49]. Older transactional data are usually excluded from OLTP due to performance requirements and are usually archived in the data warehouse. To compute the risk factor associated with an account both historical transactional data and recent transactions should be used to get a more accurate picture. In this paper, we propose an approach that computes the risk

factor of a credit card account using both archived data from the data warehouse as well as recent transactions from OLTP. In our approach, the risk probability from the recent transactions is calculated using two methods: *Standard Transaction Testing* along with *Customer Specific Testing*. To show a proof of concept, we have two different sources of data: one as offline data set and another as online dataset. Furthermore, to validate our proposed approach, we have used another dataset. We hypothesize that, our approach can be used to predict whether an account is bad or good in real time as a transaction occurs, which can then be used by a credit card company to take a more proactive action when it comes to verifying transactions and a customer's ability to pay.

1.1 Types of Fraud

A customer's inability to pay, or default on payment, or personal bankruptcy, all refer to the same thing. But the way it happens may be different. Sometimes it is due to the reason of a sudden change in the income source of the customer. Sometimes it is a deliberate process, for instance, the customer knows that he/she is not solvent enough to use a credit card anymore, but still uses it until the card is stopped by the bank. This is a kind of fraud which is very difficult to predict where there are different kinds of frauds in the financial area. The work focuses on the credit card fraud.

Financial Fraud can be defined as the intentional use of illegal methods or practices for the purpose of obtaining financial gain [19]. It is a big issue for individuals, organizations, governments, and other sectors. The rise of the internet, cloud computing, automation, and different e-payment channels is fueling this issue more even more. The most common financial fraud can be categorized as shown in Figure 1.

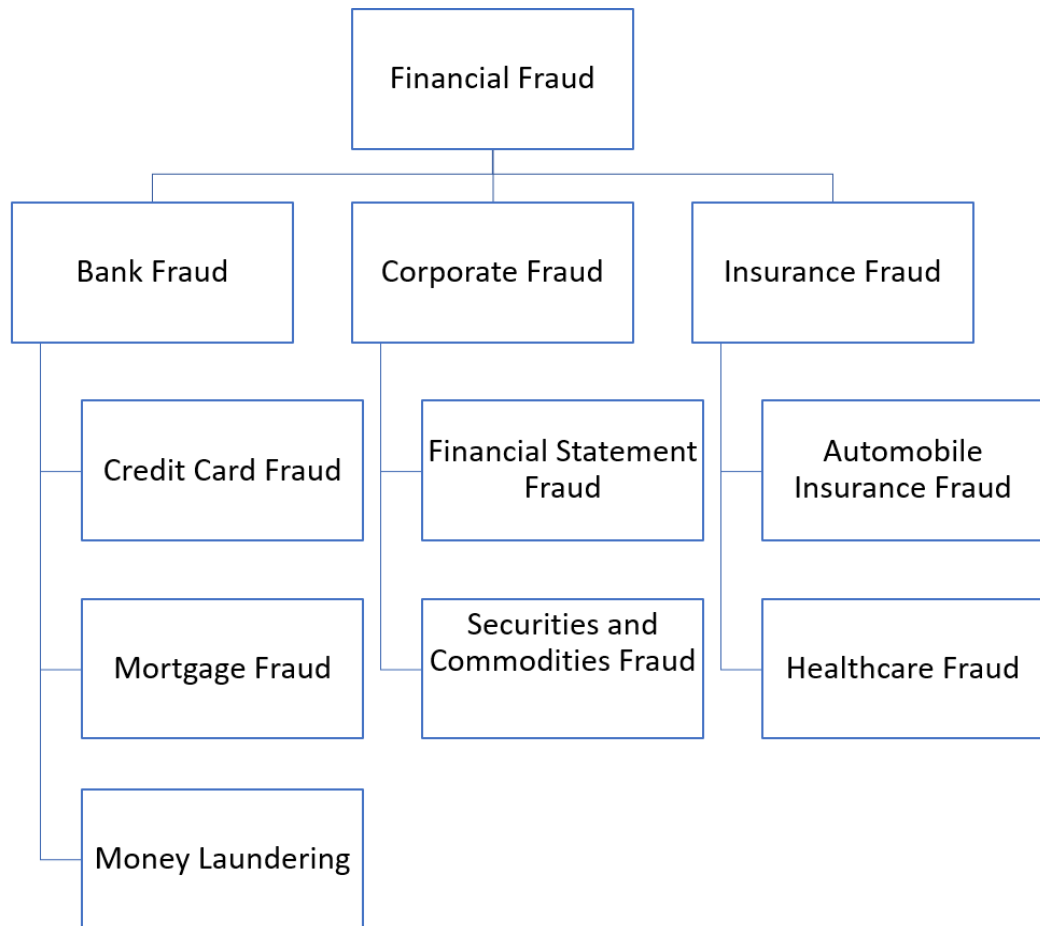


Figure 1. Categories of financial fraud

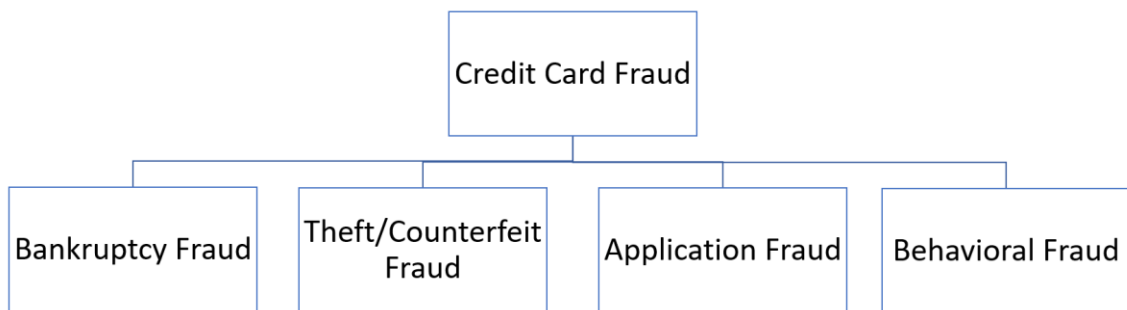


Figure 2: Most common credit card frauds

Furthermore, the most common credit card fraud can be categorized as shown in Figure 2.

According to [15], these are some of the more prominent fraud related to credit card:

- **Application Fraud:** Identity theft usually leads to application fraud. The fraudster uses real information from other people and a copy of real documents for the credit card application. There are two types of situations: a) two applications from the same individual with same detail is called duplicates (e.g., user fill up both the online application form and the credit card application form that comes with the promotional offer letter), duplicates usually are genuine, b) application from a different individual with similar detail is called identity fraudsters.
- **Theft/Counterfeit Fraud:** Fraudsters stole others credit card information and use it as much as possible times until the card is blocked by the company or the user. Fraudsters also use stolen credit card information to make a fake magnetic swipe card.
- **Bankruptcy Fraud:** It is one of the most difficult types of fraud to predict in advance. The cardholder is not solvent enough to use a credit card but still continues to spend using that card. The cardholder knows that he/she will fail to pay the final bill until the card is ceased by the bank. In the end, the cardholder will recognize him/herself in a state of personal bankruptcy which ultimately fails the bank to recover their debts.
- **Behavioral Fraud:** It happens when the details of legitimate credit card information are obtained fraudulently and the information is used to purchase goods online or over the telephone by acting as a real cardholder. This type of fraud can be detected by analyzing actual card holder's card usage patterns.

According to [33], the following are some other kinds of credit card fraud:

- **Electronic or Manual Credit Card Imprints:** Skimming is a process where the stored card information on the magnetic stripe is electronically copied on to another. For example, a pocket electronic magnetic stripe reader.
- **CNP (Card Not Present) fraud:** If the fraudsters know the card number and expiration date then it can try random 3 digit verification code for small transactions until the transaction succeeds. And if fraudsters know the 3 digit verification code then the task becomes even easier.
- **Lost and Stolen Card Fraud:** The card is out of a user possession due to either theft or loss. Anybody can do the fraudulent transaction over the internet using the physical card.

- **Card ID theft:** It happens when the criminal gets credit card information and uses that information to take over the account or open a new one or change the mailing address. It is also a difficult type of fraud to identify, and which takes longer to reveal.
- **Social Engineering:** The fraudsters try to get credit card information using a phone call or email by posing as a member of a genuine institution like financial institutions.
- **Mail non-receipt fraud:** Somehow intercept the credit card which is on the way to actual applicant's mailbox. People living in the apartment with multiple adjacent mailboxes are common victims of this type of fraud.
- **Site Cloning:** Fraudsters create an identical website of a well-known e-commerce site and deceive customer by improper use of their card information.
- **False Merchant Websites:** Offering a very cheap product or service where the main motive is to take credit card information and use it in somewhere else to do fraudulent transactions.

1.2 Challenges in Financial Fraud Detection

There are lots of challenges in the Financial Fraud detection area. Some of those challenges are as follows:

- Financial fraud is an evolving field. Need to stay ahead of perpetrators.
- Financial fraud detection methods are problem specific.
 - No fixed choice of a particular data mining approach
 - Sometimes hybrid methods work better.
 - Tuning of parameters improves results. Lots of trial and error needed to come up with an optimal set of parameters.
- Privacy issues led to reluctance in information sharing by corporations which led to different experimental limitations like undersampling.
- Financial fraud needs nearly real-time detection to avoid loss. Quick detection area of research is under-focused. For example, real-time credit card fraud detection.
- Misclassification comes with the cost of revenue and/or business. So, performance (accuracy & time) vs misclassification cost needs more focus.

- A generic framework that adapts with frauds detection of multiple domains would be valuable.
- For an electronic transaction fraud attributes like source account, destination account and amount are available but not the purpose of spending which can be termed as lack of forensic evidence. Another example is: the case of identity theft, where logs in the banking system can be found but not the whole compromised process [event logs of the user's computer, as it was attacked/compromised first].
- Financial dataset is highly imbalanced [i.e., a few fraud transactions in millions of transactions]

1.3 Limitations

Besides the challenges mentioned before, there are also limitations in the traditional approaches. Financial fraud detection is becoming more and more challenging as technology advances and the amount of data is increasing day by day. Traditional auditing process for detecting financial fraud is infeasible nowadays because it is manual, time consuming, expensive and inaccurate. Moreover, most of the empirical financial variables don't comply with traditional statistical conditions like adhering to a distribution like normal distribution.

1.4 Machine Learning is a good fit for fraud mining

Data Mining and Machine Learning techniques are a good candidate to cope with the above situation. Traditional statistics works very well with linear, repeatable, scientific analysis related to the environment where a very tight assumption is made beforehand about the data and the data distribution. It also works well with a small sample of data. But human behavior is difficult to standardize so it is difficult to fit traditional statistics in this case. Rather *Machine Learning* techniques have some advantages:

- They can work without any relationship between variables in the dataset.
- Some Machine Learning algorithms work like a black box. For instance, *Neural Nets* which is very difficult to comprehend (the internal working mechanism), but when data is passed to it, it learns from the data and applies the knowledge to the new similar dataset.

- They can work with a high volume of data (single, multiple dimension). For some algorithms, the more data the more accurate the model. Nowadays billions of devices are connected to the internet and producing tons of data. Traditional computing and algorithms are not sound enough to deal with this huge volume of data. That is the reason why the tech industries are moving towards adopting artificial intelligence.

Artificial Intelligence (AI) focuses on understanding intelligence and how to replicate the intelligence in machines (systems or agent). *Machine Learning (ML)* is a branch of *AI* which focuses on the automatic discovery of regularities in data through the use of computer algorithms and generalizing those into new but similar data [21]. Specifically, *Data Mining* and *Predictive Analysis* are an application of Machine Learning (ML), including *Fraud Detection*, *Anomaly Detection*, etc.

1.5 Some of the Applications of Machine Learning

Some of the recent and prominent applications of Machine Learning are as follows: [38]

- Facebook's face recognition system (Tagging of friends face in the photos automatically)
- Playing games without a joystick, only using motion actions (e.g., Kinect Sports). (Random Forrest algorithm is used here)
- Virtual Reality (identify movement of head and eye and then the pictures moves based on that)
- Speech to text or voice recognition.
- Dogs learn how to walk on their own (e.g., Robot dog). Reinforcement learning is used here.
- Customized Facebook ads.
- Amazon, Netflix, Audible – use machine learning for recommendation system.
- Exploring new treasure on the Mars.

Most popular machine learning algorithms are of two types.

- Supervised: Training data comprises of both the input and the desired output.
 - Classification: To which discrete class an entity belongs (e.g., whether a customer will default on payment).

- Regression: Predicting the continuous value of an entity's characteristics. (e.g., How much a customer will spend a month on the credit card, given all other available information)
 - Forecasting: Estimation of aggregated variables. (e.g., Total number of credit card fraud in a month).
 - Attribute Importance: Identifying attributes/variables that mostly affect the outcome of classification/regression. (e.g., Whether the customer owns a house or rents?)
- Unsupervised: Training data comprises of only input without any desired output. It can be used to discover hidden pattern in the data.
- Clustering: Finding natural groupings in the data.
 - Association models: finding frequent patterns, correlations, associations from the data. For example, peanut butter and jelly are often bought together.

1.6 Objective and Research Questions

Financial fraud detection is very challenging due to its dynamic nature. Moreover, it is really crucial to detect fraud as early as possible. The earlier fraud is detected, the less the loss for an individual or institution. But the prediction of fraudulent activities requires lots of data processing of recent transactions as well as historical data. The main objective of our research is to find an optimal solution to predict fraudulent or bad credit card accounts from both online and offline data in near real time. This led to the following research questions:

1. How is this research going to provide benefit to the financial institutions?
2. Is this research going to provide benefit to the customer too? If so then how?
3. Is this research going to detect individual credit card transaction fraud too?
4. Why is *Machine Learning* approaches better than traditional techniques?
5. In which aspects this approach is going to perform better than using only *Machine Learning* algorithms on the dataset?

First, our approach will help companies by identifying potential credit card bankruptcy in advanced than traditional approaches. Second, this research will provide an indirect benefit to the customer as companies can take proactive actions to detect and predict potential bankruptcy which will lead the customers to avoid a long terms debt and financial burden. Third, since all OLTP transactions are checked for possible rule violations, which contributes to risk factor determination, the proposed approach will help to detect transactional fraud. Fourth, traditional statistics works well with linear, repeatable, scientific analysis related to the environment where a very tight assumption is made beforehand about the data and data distribution and the sample size is small. But human behavior is difficult to standardize. Moreover, most of the empirical financial variables doesn't comply with the traditional statistical condition like a normal distribution. However, incorporating *Machine Learning* based techniques can be an optimal way to deal with this as it helps to deal with a huge volume and variety of data with optimal resources and time. It also helps us to get valuable knowledge from the huge volume of data without having an in-depth idea of the data and the relationship among attributes in advance. Finally, our approach selectively feeds OLTP transactions to appropriate methods where a transaction is never used more than once, and the calculated risk factor is carried forward for the evaluation of future transactions. Many risky accounts will be detected in the very early stage, which saves revenue. In addition, while some risky accounts cannot be detected using only machine learning algorithms, but with our approach of standard and customer specific rules, the recall will be improved a lot with a minimal increase in precision.

1.7 Contributions

The main contributions of this research are as follows:

- An approach for predicting credit card bankruptcy fraud from both historical and transactional data simultaneously in near real time.
- An optimized way to avoid loss (by detecting fraud in advance) by efficiently using *Machine Learning* approaches.
- Reducing redundant processing by selectively feeding an optimal set of data to the appropriate algorithms.

- Integrating an *Extremely Randomized Trees* algorithm in our approach which performs better than the state of art results on a given dataset.

1.8 Organization

The rest of the work is organized as follows: in Chapter 2 background information and literature review on different types of financial fraud with the main focus on credit card fraud and default prediction is presented; in Chapter 3 our proposed approach is presented along with the research methodologies; in Chapter 4 experiment setup used for the tests and validation techniques are presented; and in Chapter 5 the results of the experiments are discussed and analyzed; and in Chapter 6, we presented the summary of this work along with some future directions.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

2.1 Machine Learning Algorithms

We have found different types of *Machine Learning* algorithms are used for different fraud detection. Some of the algorithms are supervised (Training data comprises of both the input and the desired output) and some of the algorithms are unsupervised (Training data comprises of only input without any desired output). Furthermore, supervised algorithms have different types such as: classification, regression, forecasting, attribute importance, and anomaly detection. On the other hand, unsupervised algorithms can be of types such as: reinforcement learning, clustering, association model etc.

2.1.1 Decision Tree Classification:

It is helpful to predict categorical values or outcomes. For example, we want to predict a person is going to play golf or not based on the weather condition. *Decision Tree* algorithm breaks down a dataset into smaller subsets and builds tree incrementally. It uses *entropy* or *information gain* to select the attribute to divide upon. ID3 is the core algorithm to build a decision tree. Entropy is the randomness in the dataset. If all the samples are homogeneous in nature then there is no randomness in data, that means the entropy is zero [40]. On the other hand, information gain is the decrease in the entropy after a division of the dataset based on some attribute. The more information gains the more important the attribute. At each step of the decision tree algorithm, an attribute is selected as the splitting node based on the information gain. Entropy and Information Gain calculations are based on the frequency count of instances.

2.1.2 Decision Tree Regression

It helps with predicting a decision that can have continuous value (e.g., 75%) rather than just the discrete values (i.e., yes/no). For example, we want to predict the golf played hours instead of just the decision played or not. In this algorithm, the same ID3 algorithm is used but here *Standard Deviation Reduction* is used instead of *Information Gain*. Here standard deviation is used to calculate the

homogeneity of samples [40]. If all samples are completely homogenous then the standard deviation is zero. So, after a split, the attribute with highest *Standard Deviation* reduction is treated as next splitting attribute.

2.1.3 Linear Regression

When the dependent variable or expected output (i.e., airfare cost) is continuous valued then the linear equation works better. The equation: $y = b_0 + b_1 * x$ is a linear equation where y or *airfare cost* (from figure 3) is a dependent variable which is dependent on x (distance to the destination from Figure 3). Usually the more distance, the more cost. b_0 is a constant, here suppose it is the starting/minimum fare for any distance travel. b_1 is the slope or increase of cost per mile. From the Figure 3, we can see that a line is drawn for the machine learning model. Basically, a lot of lines are drawn and the line that has lowest value [ordinary least squares] of the sum $(y - \hat{y})^2$ is chosen as the machine learning model. When a new sample is added then the distance to destination in the mile is observed and the corresponding y value on the line is chosen as the airfare cost for the new sample.

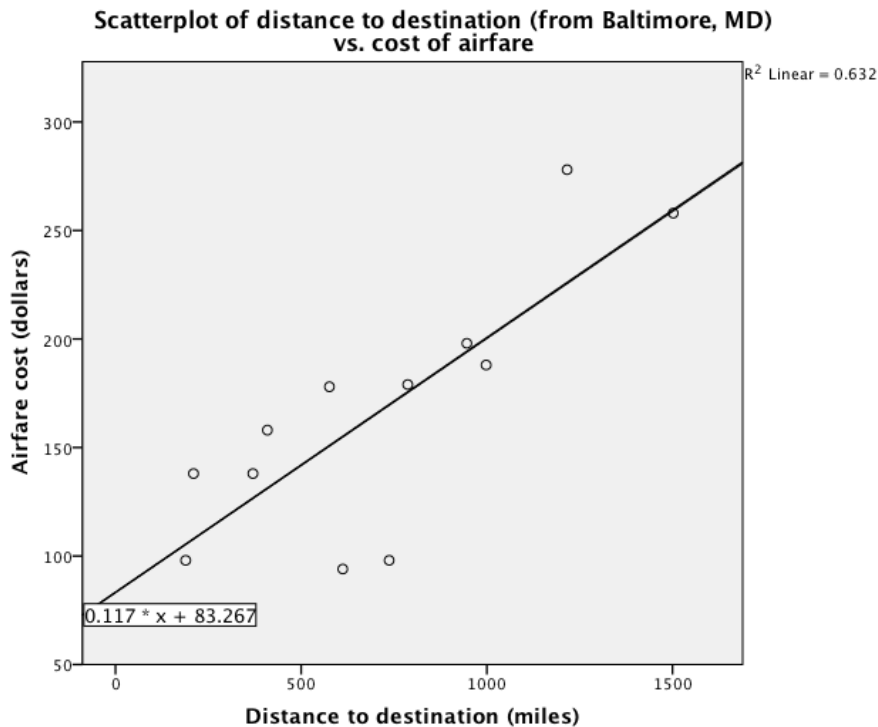


Figure 3: Linear Regression [41][42]

2.1.4 Logistic Regression

When the dependent variable or output is categorical (yes/no) then the logistic regression $[Y = (e^X) / (1 + e^{-X})]$ works better. But when more than one independent variable is used to predict the value of a dependent variable then **multiple linear regression** $[Y = \beta_0 + \beta_1 X_1 + \dots + \beta_k X_k + \epsilon]$ is used. Here ϵ is the error or residuals.

2.1.5 Support Vector Machine

Support Vector Machine algorithm tries to separate the objects with a line in between that has the maximum margin. From the Figure 4, we can see that we can draw multiple lines in between for separating objects of different categories but we need to take the line that has the maximum margin or separation. The two points on the dotted line are called support vectors.

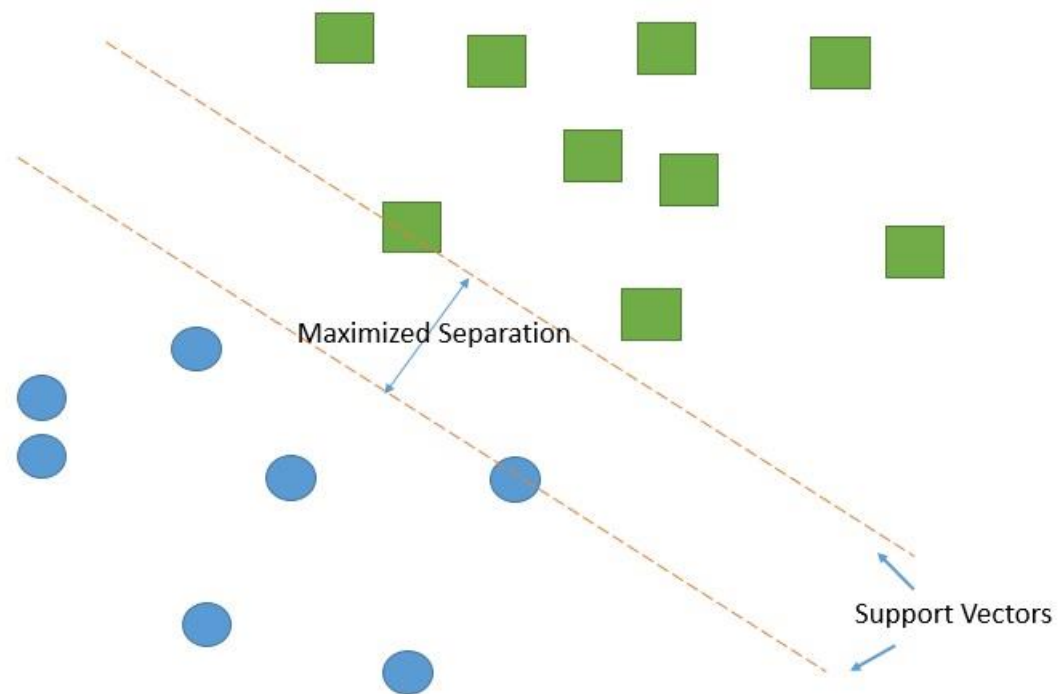


Figure 4: Support Vector Machine [42]

2.1.6 Fuzzy logic based system

It is based on the concept that whether to perform an action or not to perform an action. Also, the intensity of the action is based on a probability, not a binary value. For instance, in traditional logic, whether to break the car or not for a situation is the binary decision (if the car is close enough to another car then 1 otherwise 0). On the other hand, according to the *fuzzy logic*, the decision is not binary rather it has some continuous probability value. It gives a more detailed decision like how close or far is the car ahead.

2.1.7 Naïve Bayes

Naïve Bayes classifier is a probabilistic classifier based on the Bayes Theorem with the strong assumption of independence among features. Here the independence of variable refers to the matter that it doesn't assume any correlation among the features. Each feature has a probability which contributes to the final outcome independently.

2.1.8 Random Forest

Random Forest (RF) is an ensemble technique where multiple models are used to produce the final prediction. In Random Forest, multiple algorithms (same or different) are used to produce a more accurate result. Usually, the dataset is divided into a different random subset. A subset may have different random attributes too. For each of the subset of data, a decision tree is generated. These trees are not correlated. When a new instance is found then the class of that instance is calculated using all the trees. Later a vote count is done, the class with the highest vote is assumed as the predicted class for that new instance.

2.1.9 Extremely Random Trees

In Extremely Random Trees (ET) randomness goes further than the Random Forest. In Random Forest the splitting attribute is determined by some criteria where the attribute is the best to split on that level, whereas ET splits nodes by choosing random cut-points. Moreover, ET applies the entire training set to train the tree instead of using bagging to produce the training set as in Random Forest. Sometimes, ET

gives a better result than Random Forest for a particular set of problem. The cut-point or threshold randomization reduces the variance at the expense of little bit increase of bias.

2.1.10 Artificial Neural Network

In the Artificial Neural Network (ANN), the information is processed imitating the working procedure of the neuron of our memory. In Figure 5, each circle (i.e., X_1) on the left represents an attribute (independent variables) which has an initial information and weight in the path between the node and the node in next layer. Here we are showing only one layer but actually, there may be multiple hidden layers in between. Then the circle in the middle has the activation function which chooses one of the activation function and applies that to produce the output value (Y). After an iteration, there are multiple options for calculating the difference between the actual output and the output from the ANN. To reduce the gap, the weight is adjusted through a back propagation procedure until it reaches a satisfactory point. By this way, we get an ANN which is optimal for the training data set and which can be applied to a new dataset.

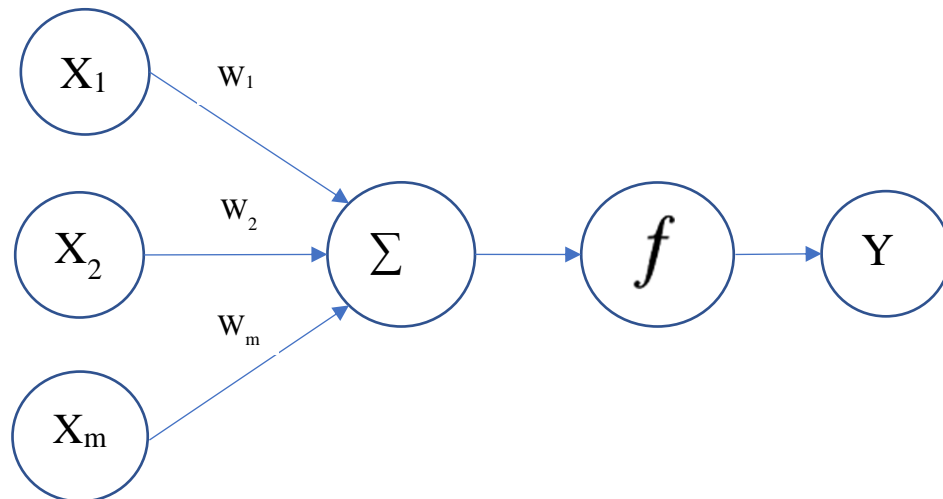


Figure 5: Artificial Neural Network

2.1.11 Deep Learning

A simple Artificial Neural Network (ANN) usually have zero or one hidden layer. But when there are lots of hidden layers in between input and output layer then the networks become more complex and then the learning process of the network is called deep learning.

2.1.12 K-means clustering

Initially, a reasonable number of cluster is assumed. Then for each cluster, a centroid is assumed. There are some algorithms to calculate a reasonable number of the cluster at the beginning. Distance from each point to all other centroid is measured and each point is assigned a centroid that has a minimum distance. After that, the centroid is adjusted based on the average distance to other points. This process continues until no more centroid adjustment found.

2.1.13 Principal Component Analysis

Principal Component Analysis (PCA) extracts important variables from a large set of variables in the dataset. The main motive of PCA is to capture as much as information with an optimal number of variable or dimension and with keeping data integrity [43]. Dimensions reduction helps in discovering hidden pattern or correlation, removing redundant or noisy data, efficient data storage and processing, and in better interpretation and visualization.

2.1.14 Hidden Markov Models

Hidden Markov Model (HMM) [44] is good for temporal data analysis like voice recognition, handwriting recognition. HMM has multiple states of the model. Each state has a probability value from moving into another state. And an observation can be originated from multiple states (many to one relation). For example, some speakers sound is coming into our ear. From this sound, we can't actually figure out the state of the system that generates the sound. Instead, the true state of the system is the collection of parameters (frequency of vocal chords, the semantic meaning behind the sound, shape of a speaker's mouth) that determine which sound to generate. Moreover, a different combination of these

parameters can converge into same observation so we can't tell the actual state of the system from the observation, instead the actual nature is hidden.

HMM has two rules as follows:

1. Markov property: System moves from current state to next state (may be the same state) based on some probability distribution that is only dependent on the current state.
2. Hidden state: After each transition, the model emits an observation whose distribution is only dependent on current state and the observer of the observation cannot exactly tell which states produce the observation. These unknown states those produce the observations are called hidden states.

2.1.15 Genetic Algorithm (GA)

It is an algorithm that mimics the biological evolution process. The algorithm starts with a random initial population (genes or instances). After that, the algorithm runs a number of iteration/generation until a stopping condition (time limit, number of generations, fitness limit, tolerance, etc.) is met. At each generation/iteration, the algorithm does following steps: [28]

1. Score a fitness value to all individual members of the current population (genes or instances).
2. Categorized instances into elite (individuals with best fitness values, those are directly passed to next generations) and parents (individuals with better fitness values).
3. From a parent, it creates children for next generation by following:
 - a. Mutation children: by introducing random changes or mutation into a single parent.
 - b. Crossover children: by combining properties/vectors of a pair of the parent.
4. Replace current generation with children and elite instances to form next generation.

GA algorithm kind of expensive though it is faster than brute force search. It performs well to solve the combinational problem to solve part of the problem efficiently. GA algorithm always could not provide an optimal solution but it finds the best solution to a problem efficient within the stopping conditions.

2.1.16 Meta-Learning

Meta-Learning focuses on the relationship between task/domain and the learning strategies. The end users often don't understand which model is most suitable or which set of models can be taken to proceed with trial an error basis for a particular problem. Meta-learning system provides automatic and systematic guidance to the user by mapping a particular task into the most suitable model or combination of models [29]. The meta-learning process has two modes, the first mode is called *knowledge acquisition mode*. The goal of this mode is to learn about the learning process. It starts with extracting characteristics (i.e., meta-features) from the dataset/datasets. In the *learning techniques*, each dataset is considered independently and knowledge is gathered. The output from the learning techniques is the final learning strategy which may be a classifier or combination of classifiers. Performance of the learning strategy is evaluated and combined with the information derived from the *Meta Feature Generator* to form a combined meta-knowledge-base. In the next mode (Advisory mode), the meta-features extracted from a new dataset are matched with the meta-knowledge base to produce the recommendation for the best available learning strategy. Meta-Learning is flexible: it is possible to select, alter, and combine different algorithm to solve a problem effectively. It also provides the scope of improving the learning process over time. The main disadvantages are the limited number of samples for meta attributes (usually at the beginning) and the problem of metadata metric (e.g., total sales to revenue, etc.).

2.1.17 Comparison of ML algorithms

In general, unsupervised classification has low accuracy. An unsupervised approach like hidden Markov model is used to detect outlier or spike when the data set is unlabeled. Fraud analysis or misuse detection is usually supervised (rule induction, decision trees, neural network), on the other hand, anomaly detection or user behavior analysis is unsupervised. Table 1 shows advantage and disadvantage of different algorithms used in fraud detection. This table is a compilation of our literature survey and from the research of [39] and [16].

Table 1. Advantage and disadvantage of different algorithms

Technique	Advantages	Disadvantages
Genetic Algorithm	It is efficient to the large combinational problem [faster than brute force] as it can be combined with other problem to increase the performance. Works well with noisy data. GA algorithm always could not provide an optimal solution but it finds the best solution to a problem that is efficient within the stopping conditions.	Computationally expensive, though faster than brute force. It is undirected search, difficult to direct into an optimal solution (if known). Difficult to understand and requires extensive knowledge to set up the tools.
Meta-Learning	Flexibility: It is possible to select, alter, combine different algorithm to solve a problem effectively. Inductive Transfer: learning process improves over time.	Limited number of samples for meta attributes, problem of metadata metric (total sales to revenue, etc.)
Artificial Immune System (AIS)	Good in pattern recognition	High training time needed. Doesn't work well with noisy data.
Artificial Neural Network (ANN)	Good for abstract or complex problems (ex. Image recognition). Scales well for a larger data set with GPU and CUDA software. Can significantly outperform other models if conditions are right.	Not easy to comprehend (poor explanation capability). Nonnumerical data need to be normalized. High training and processing time for the larger neural network. Don't perform well on a small dataset, the Bayesian or any linear approach has an advantage here.
Hidden Markov Model (HMM)	If there is unobserved variables or states (for example high or low volatile period) then it works better. It is fairly readable	Expensive in terms of memory and computation time. No way to

	and explainable statistical model with a strong foundation. It has a wide variety of applications classification, pattern recognition etc.	express dependencies between different hidden states.
Bayesian Network	It is robust and safe. Good for simple interpolation.	The learning process is computationally expensive. Perform poor on high dimensional data. Also, it is hard to interpret.
Support Vector Machine (SVM)	It is robust compared to linear regression, works well even with bias data. Better generalization or less overfitting. Works well with fewer training samples too.	Doesn't work well with a large dataset. Training the model takes comparatively long time.
Fuzzy logic	Simple and user-friendly, easy to understand and implement. Efficient performance.	Requires lots of tuning and simulation before it is in operation. Requires presentation of knowledge in <i>if then else</i> format.
Expert System	The system works systematically, doesn't jump to a conclusion by omitting something. Works well with very large datasets and data never deleted rather it keeps accumulating data.	Doesn't work well with missing values. Performs poor in the integration.
Decision tree (DT)	Easy to understand, and explain. Internally do feature selection, need little effort for data preparation.	It can be easily overfitted. Though tree pruning can help to negate that. Need to check each condition one by one. A slight change in data can give a completely different tree.

2.2 Fraud

While there are many types of fraud, the focus of this research is on bad credit card accounts.

2.2.1 Financial Statement Fraud

A financial statement represents the financial status of a company. It helps in decision making for investor, creditors, and managers. There are lots of financial statement frauds visible in recent years. According to the US Committee of Sponsoring Organizations of the Treadway Commission (COSO), a *fraudulent financial statement* is a conduct either intentional or reckless based on false information or omission that results in significantly misleading financial reports. Some of the examples include Enron case in 2001 and the WorldCom case in 2002 in the United States, the Infodisc and Summit Technology cases in 2004 in Taiwan. The cost of the fraudulent financial report in the United States is estimated to be billions of dollars in each year [18].

In the work of [3], [7] and [9] different types of Financial Statement Frauds (FFS) are discussed. In the work of [3], the authors explored the effectiveness of different Data Mining (DM) classification techniques for detecting Financial Statement Fraud (FFS). They also identified the key factors associated with FFS. The classification methods used by the authors are Decision Trees, Neural Networks and Bayesian Belief Networks (BBN). The input for their experiments are different ratios and variables like Total Assets, Working Capitals, Fixed Asset to Total Assets, Sales to Total Assets, Net Income, Quick Assets, Liabilities, Earning before interest and taxes, those are derived from the financial statement. They also focused on the management fraud which is induced by the managers to fulfill the target and hiding losses or debt. According to them, financial distress is also a motivation for the management fraud. To accommodate financial distress in the experiment they have used the well-known Altman's Z-score [20][17] which is used in a lot of research relevant to bankruptcy prediction and financial distress calculation. They found that Bayesian Belief Network outperforms the other two models in terms of classification accuracy. Neural Networks achieve satisfactory high performance, on the other hand, the result from the decision tree is relatively low. To get a more accurate result and reducing models bias they did 10-fold cross validation which is a technique where the whole dataset is divided into 10 equal subsets.

In each iteration (total 10 iterations) one of the subsets is assumed as data to be tested and remaining 9 subsets are assumed as the data to train the model. In the end, results from all the iterations are averaged to get the final result. Due to the software limitations that they have used in the experiment, they used discretized data instead of continuous data. According to the authors, data discretization helps to eliminate the effects of outliers though they are not sure how much it affects the model performance. Further research is needed for assessing that effect on performance. Their research can be a great assistance to the auditors, tax authorities, credit scoring agencies, stock exchange and the law firms for assessing FFS.

In the work of [7], they proposed a valid and rigorous financial statement detection model. Their research objects are companies which experienced both FFS and non-FFS cases between the years 2002 and 2013. They have used Artificial Neural Network (ANN), Support Vector Machine (SVM), Bayesian Belief Network (BBN) and Decision Tree (DT) to detect FFS. The conventional regression analysis or statistical approaches have high error rate compared to these data mining approaches. According to [19], empirical financial variables often cannot comply with relevant statistical conditions like a normal distribution. So, machine learning approaches have emerged to identify FFS as it doesn't require statistical hypotheses of data combinations. In this research, they used total 30 financial and non-financial variables. Some of the financial variables include the current asset to total asset ratio, net income to total asset ratio, gross profit to net sales ratio, cost of goods sold to average inventory ratio, pre-tax profit to net sales ratio etc. Some of the non-financial variables include the size of board directors, the ratio of stocks held by directors and supervisors, number of outside supervisors. They used Decision Tree (DT) to select important and representative variables. From their experiment, they found the accuracy of DT CHAID (Chi-squared automatic interaction detector) is relatively high and can be used as a tool to help detection of FFS.

In the work of [9], the authors provided a comprehensive study Financial Fraud Detection (FFD) process. They mainly focused on Financial Statement Fraud (FSF). They applied Regression, Neural Network, Bayesian Tree and Support Vector Machine to detect fraudulent FFS. The financial variables they have used and found as the important contributor are as follows: a) the inventories to sales ratio b) total debt to total asset ratio c) net profit to total asset ratio d) financial distress (z-score). And they conclude that companies with high inventories with respect to sales, high debt to total assets, low net profit to total asset, low working capital to total asset and low z-scores are more likely to falsify financial statements.

In the work of [6], the author's categorized different types of financial fraud as follows: 1) Banking fraud 2) Corporate fraud 3) Insurance fraud. They further divided banking fraud into three sub-categories as follows: credit card fraud, mortgage fraud, money laundering. They also divide corporate fraud as financial statement fraud and security and commodities fraud. And they divide the insurance fraud as automobile insurance fraud and health care fraud. They mentioned that the traditional auditing based fraud detection is not feasible in the age of big data. Even the traditional statistical based approach is not perfect for most of the cases. Rather, current computational intelligence based data mining approaches like SVM, ANN is more appropriate to solve recent fraud cases. They mentioned different challenges in the financial fraud detection area. Some of those are as follows: undersampling of data due to lack of financial data set, real-time fraud detection requires huge computational performances, fraudsters change their techniques frequently to remain undetected, misclassification cost, lack of a generic framework that can be applied to multiple fraud categories etc.

2.2.2 Credit Card Fraud

The research work of [8], [12], [14], [15], [16], [23], [24], [35], [36], and [32] are all about credit card fraud detection and prediction using different knowledge discovery approaches. In the work of [14], they have used a genetic algorithm to detect credit card fraud. They focused on the categorized credit card fraud such as monitorable fraud, critical fraud, and ordinary fraud. They have accumulated fraud weight for each of the criteria applicable to a particular account. The criteria are credit card usage frequency, location, credit card overdraft, credit card book balance, Avg daily spending. Each criterion has different weight factors too. They also advocate reducing the total amount of fraud rather than only focusing on reducing the number of fraud.

In the work of [15], the authors mentioned that fraud can occur with any credit products like a credit card, personal loans, home loans and retail. But the credit card is the most famous target of fraud. They furthermore divide fraud into following types:

1. Bankruptcy fraud: the customer knows that he/she will be unable to pay the bill for purchases still using the card. In the end, the customer recognizes itself in a state of personal bankruptcy and not able to pay the debt. Bankruptcy fraud is one of the most difficult types of fraud to predict.

2. Theft fraud/ counterfeit fraud: Theft fraud done is using a card that is not yours and using it without the customer's permission. The fraudsters steal the card and use it as many as times possible until the card is blocked by either the company or the customer. Counterfeit fraud occurs when the card is used remotely without the permission of the cardholder; in this case, only credit card details are needed. No physical card or signature is needed.
3. Application fraud: applying for a credit card with false information. Two scenarios: two application from the same applicant with same details which is called duplicates, two application with similar information but from two different applicants which is called identity fraudsters.
4. Behavioral fraud: this type of fraud occurs when the detail of legitimate cards have been obtained fraudulently and transactions are made with the card details.

In the work of [8], the authors used an *ontology graph* based credit card fraud detection approach. In a *knowledge base* graph, types, properties, and the relationship among entities are mapped with real values. For example, entities are represented as nodes and relationship among them is represented as edges, here nodes and edges presented real values, not with any abstract terms. On the other hand in an ontology-based graph, the entity types, properties and the relationship among the entities are formally described. It is not expressed with real values. It is more formal and abstract. Here the authors generated 5000 transactions by following the data structure and did the experiment on this dataset using Matlab. When a new transaction occurred the ontology graph is generated for that transaction, and then the system looks for similar ontology [in terms of pattern] in the database and fetch similar ontology graphs. After that, it is checked whether the distance between recent transactions ontology graph and previously stored transaction ontology graphs is within the accepted threshold, if not then it is assumed as a fraudulent transaction.

2.2.3 Bankruptcy/Default on Payment Fraud

The research work of [1], [2],[5], and [13] are all about personal bankruptcy or credit card default on payment prediction and detection. In the work of [5], the authors worked on finding financial distress from four different summarized credit datasets. Bankruptcy prediction and credit scoring were their main aspect of the financial distress prediction. According to the authors, a single classifier is not good enough for a

classification problem of this type. So, they suggested using ensemble approach where multiple classifiers are used on the same problem and then the result from all classifiers are combined to get the final result. This helps in reducing Type I/II error. Lowering type II error or false positive is very crucial in the financial sector. For classification ensemble they have used four approaches as follows: a) Majority voting b) Bagging c) Boosting and 3) Stacking. They also introduced a new approach named *Unanimous Voting (UV)* where if any of the classifiers says yes then it is assumed as *yes* whereas in *Majority Voting (MV)* at least $(n+1)/2$ classifier needs to say *yes* to make the final prediction *yes*. Though this reduces the Type II error but decreases the overall accuracy.

In the work of [13], the authors present a system to predict personal bankruptcy by mining credit card data. In their application, each original attribute is transformed either to a binary [good behavior and bad behavior] categorical attribute or multivalued ordinal [good behavior and graded bad behavior] attribute. Consequently, they obtain two types of sequences, i.e., binary sequences and ordinal sequences. Later they resort to a clustering technique for discovering useful patterns that can help them to identify bad accounts from good accounts. Their system performs well, however, they only use single data sources, whereas the bankruptcy prediction systems of credit bureaus use multiple data sources related to creditworthiness.

In the work of [1], they compared the accuracy of different data mining techniques for predicting the credit card defaulters. The dataset used in this research is from UCI machine learning repository which is based on Taiwan's credit card clients default cases [34]. This dataset has 30000 instances, and 6626 (22.1%) of these records are default cases. There are 23 features in this dataset. Some of the features include credit limit, gender, marital status, last 6 months bills, last 6 months payments etc. These are labeled data and labeled with 0 (refer to nondefault) or 1 (refers to default). From the experiment, based on the area ratio on the validation data they ranked the algorithms as follows: artificial neural network, classification trees, naïve Bayesian classifiers, K-nearest neighbor classifiers, logistic regression, and discriminant analysis. To get the actual probability of default rather than just the discrete binary result they proposed a novel approach, called Sorting Smoothing Method (SSM).

In the work of [2], the authors use the same dataset as of [1]. But they applied a different set of algorithms and approaches. In this research, they propose an application of online learning for a credit card

default detection system that achieves real-time model tuning with minimal efforts for computations. They mentioned that most of the available techniques in this area are based on offline machine learning techniques. Their work is the first work in this area that is capable of updating model based on the new data in real time. On the other hand, traditional algorithms require retraining the model if there is some new data. This is a big problem if the data size is big in terms of computation time, storage and processing systems. For the purpose of real-time model updating, they use Online Sequential Extreme Learning Machine (OS-ELM) and Online Adaptive Boosting (Online AdaBoost) methods in their experiment. They compared the results from above mentioned two algorithms with basic ELM and AdaBoost in terms of training efficiency and testing accuracy. In online AdaBoost, the weight for each weak learner and the weight for the new data is updated based on the error rate found in each of the iterations. The OS-ELM is based on basic ELM which is formed from a single layer feedforward network. Along with these algorithms, they also applied some other classic algorithms such as KNN, SVM, RF, and NB. Although KNN, SVM, and RF have shown highest accuracy, the training time was more than 100 times compared to other algorithms. They found RF exhibits great performance in terms of efficiency and accuracy. After all, both the online ELM and AdaBoost maintain the accuracy level of other offline algorithms, while significantly reduce the training time with an improvement of 99% percent. They conclude that the online AdaBoost has the best computational efficiency, and the offline or classic RF has best predictive accuracy. In other words, Online AdaBoost balances relatively better than offline or classic RF between computational accuracy and computational speed. They mentioned two future directions of this research as follows: a) incorporating concept drift to deal with the change of new data distribution over time which may affect the effectiveness of the online learning model b) sustaining the robustness of online learning for a dataset with missing records or noise. They also mention that some other online learning techniques like Adaptive Bagging could be applied and compared in terms of speed, accuracy, stability, and robustness.

2.2.4 Other Fraud

In the work of [4],[10], [11], and [12] different types of transactional frauds such as banking transactions, e-transactions are discussed. The research work of [10] is a dynamic model and mechanism to discover fraud detection system limitations while existing fraud detections systems use some predefined

rules and scenarios or static models. In this instance, their dynamic model updates rules periodically [10]. They use a KDA clustering model which is a combination of three clustering algorithms, k-means, DBSCAN and the Agglomerative clustering algorithm, that are then represented together as a dynamic solution[10]. However, with this approach, the accuracy obtained by KDA modeling for online data is much less than that of the offline data. In the work of [11], the authors discuss different methods on fraud detection based on decision trees using Gini impurity, information gain, and a binary decision diagram. In the work presented in [12], a data mining approach is presented using transaction patterns for credit card fraud detection where the spending pattern may change anytime due to changes in income and preferences.

In the work of[4], the authors have done a good comprehensive study of different types of e-frauds. The study was done based on the information of different e-channel situated in Nigeria. They defined e-fraud as electronic banking trickery and deception that affect individuals, business, society, and governments. They mentioned following reasons that fuel e-fraud in Nigeria: a) dissatisfied staff b) Increased e-payment system for transactions c) different emerging payment product adopted by Nigerian banks d) complexity of e-channel systems e) abundance of malicious code, tools, malware available to attackers f) rapid pace of innovations in technological areas g) lack of knowledge and proper security practices h) obscurity of internet i) dominant role of third party processors in switching e-payment transactions i) lack of active approach in fraud detection and prevention j) lack of inter-industry (banks, telecom, police, etc) collaboration in fraud prevention.

The authors of [4] also mentioned several techniques that cyber attackers use to do the fraud are as follows: a) cross-channel fraud: customer information gathered through one channel is used in another channel. For example, credit card information obtained in customer care center is used to do the fraud in online shopping. b) Data theft: hackers get access to different sites and sell confidential data, c) Email spoofing: deception through email by changing header of email so that it seems the email came from a trusted source, d) Phishing: stealing confidential information through the process of spoofing, e) smishing: deception through SMS, f) vishing: soliciting personal information by phone call to the victim, f) shoulder surfing: looking over someone's shoulder to get password, pin etc, g) social engineering: using social network to get public information to do further fraudulent activity, h) key logger: use of key logger software to get password, pin etc. i) sniffing: network packet analysis to get sensitive information, j)

session hijacking: stealing communication session to steal data. K) Man-in-the-middle attack: attacker secretly relays and alters communication messages between two communicating parties in a way that both the communicating parties believe that they are talking to themselves. But the actual scenario is: they both individually talking to the attacker. Fraud analysis or misuse detection is usually supervised (rule induction, decision trees, neural network), on the other hand, anomaly detection or user behavior analysis is unsupervised. The authors described some common machine learning algorithms briefly. They did some experiments using free and open source sophisticated statistical software package R. They used a software called *Rattle* which is based on R. Rattle provides a nice graphical user interface for performing different data mining and machine learning tasks. They mentioned different data reduction techniques as follows:

- Data aggregation: Data is collected from different or similar sources, some aggregation function is applied to make a dataset with an optimal number of variables/features for efficient data analysis.
- Attribute subset selection: discarding irrelevant, weak, redundant attributes/dimensions.
- Numeric reduction: reduce data volume by smaller or alternative form of data representation. In the parametric method, it only stores parameters instead of original data, for example, linear regression. In the non-parametric method, it does not assume any models. For example histograms, clustering, sampling.
- Discretization and concept hierarchy generation: the real value of data attributes are replaced with ranges or higher conceptual levels.
- Principal Component Analysis: It extracts important variables from a large set of variables in the dataset. Its motive is to capture as much as information with an optimal number of variables or dimension and by keeping the data integrity.

They did their experiment on a dataset of 8,641 observations with 9 attributes. They found 6 of the observations are fraudulent. The main objective of their study was to find out the best solution either single or integrated to control the fraud. They conclude that if computation time is not a big issue then nearest-neighbor based approach is better. And if computation time is a big issue then clustering based anomaly detection is good.

2.2.5 Research by fraud type

Table 2 summarized different research in tabular format according to the fraud type discussed. The fraud types are credit card bankruptcy/default fraud, credit card fraud, financial statement fraud, and banking transactions fraud.

2.2.6 Research by datasets

Different types of the dataset are used in different fraud detection research. Some of those datasets are public, some of them are private. Most of the cases, the dataset is private or not published for privacy issues. Table 3 follows list different dataset used in the research with the nature of the dataset.

Table 2. Research by fraud type

Reference	Authors	Fraud Type
[1]	Chu H et al.	Credit card bankruptcy/default accounts
[2]	Lu H et al.	Credit card bankruptcy/default accounts
[3]	Kirkos E et al.	Financial Statement Fraud
[5]	Liang et al.	Credit card bankruptcy/default accounts
[7]	Chen S et al.	Financial Statement Fraud
[8]	Ramaki A et al.	Credit card fraud
[9]	G.Appraao et al.	Financial Statement Fraud
[10]	M. Vadoodparast et al.	Banking Transaction Fraud
[12]	Lee C et al.	Credit Card Fraud
[13]	Xiong T et al.	Credit card bankruptcy/default accounts
[14]	K RamaKalyani et al.	Credit Card Fraud
[15]	Delamaire L et al.	Credit Card Fraud
[16]	Al-Khatib A et al.	Credit Card Fraud
[23]	Pun Joseph	Credit Card Fraud
[24]	West J et al.	Credit Card Fraud
[35]	Bhattacharyya S et al.	Credit Card Fraud
[36]	Gadi MFA et al.	Credit Card Fraud
[37]	Mahmoudi N et al.	Credit Card Fraud

Table 3. Research by datasets

Reference	Authors	Dataset	Dataset Type
[1]	Chu H et al.	Default payments in Taiwan [34]	Public
[2]	Lu H et al.	Default payments in Taiwan [34]	Public
[3]	Kirkos E et al.	A real dataset for 38 financial and 38 non-financial companies	Not published
[5]	Liang et al.	Taiwan bankruptcy, China bankruptcy, Australian credit, German credit	Public
[7]	Chen S et al.	Some of the companies with fraudulent and non-fraudulent FFS between the years 2002 and 2013.	Unknown
[8]	Ramaki A et al.	Artificial Dataset, 5000 records	Not published
[10]	M. Vadoodparast et al.	A real dataset with 3609618 records	Private
[13]	Xiong T et al.	Real credit card dataset	Not published
[14]	K RamaKalyani et al.	Synthetic	Not published
[23]	Pun Joseph	Real (Canadian Organization)	Not published
[24]	West J et al.	2009 UCSD-FICO dataset	Available in some cloud storage but not officially
[35]	Bhattacharyya S et al.	Real credit card transactions dataset	Not published
[36]	Gadi MFA et al.	Dataset from a Brazilian card issuer	Not published
[37]	Mahmoudi N et al.	Dataset from an anonymous bank of Turkey	Not published

2.2.7 Research by Algorithms

A different set of algorithms are used in different research. Sometimes the evaluation criteria are also different. Table 4 is the list of different research and the algorithms used. We tried to list down the best algorithm in each research with the evaluation criteria they have used too.

Table 4. Research by algorithms

Reference	Authors	Techniques Used	Best Algorithm	Evaluation Criteria
[1]	Chu H et al.	KNN, Logistic Regression, Discriminant Analysis, Naïve Bayesian, Artificial Neural Networks, Classification Trees	ANN	Error rate, Area Ratio
[2]	Lu H et al.	Online and offline Extreme Learning Machine, Online and offline AdaBoost, KNN, SVM, RF	Online AdaBoost	Accuracy and Time
[3]	Kirkos E et al.	DT, NN, BBN	Bayesian Belief Network	Accuracy
[5]	Liang et al.	Single classifier: SVM, KNN, CART, MLP. Classifier ensembles: Bagging, Boosting, Stacking, Majority Voting.	Majority Voting (Classifier Ensembles)	Accuracy, Type I error, Type II error
[7]	Chen S et al.	DT, BBN, SVM, ANN, CHAID-CART	CHAID-CART	Accuracy
[8]	Ramaki A et al.	Ontology Graph		Accuracy
[10]	M. Vadoodparast et al.	K-means, DBSCAN, Agglomerative	KDA (A dynamic model, a combination of 3 algorithms)	TPR, FPR, TNR, FNR
[13]	Xiong T et al.	K-means, SVM	SVM	ROC
[14]	K RamaKalyani et al.	Genetic Algorithm	GA	1. Based on CC Over Draft 2. Based on CC Book Balance 3. Based on CC usage Location.
[23]	Pun Joseph	Meta-Learning Strategy		TPR, FPR, ROC, F1

[24]	West J et al.	Genetic Algorithm Neural Network, SVM, Random Forest, Fuzzy rule	SVM	Accuracy, Sensitivity, Specificity, Precision, FPR, F-measures, F2
[35]	Bhattacharyya S et al.	Random Forests, SVM	Random Forests	Accuracy
[36]	Gadi MFA et al.	Artificial Immune Systems (AIS), Neural Nets(NN), Bayesian Nets (BN), Naïve Bayes (NB), Decision Trees (DT)	AIS	A cost function consists of FN, FP, and TP.
[37]	Mahmoudi N et al.	Fisher Linear Discriminant Analysis (FDA), Modified FDA, ANN, DT, NB	FDA and MFDA	Profit and Time

2.2.8 List of attributes found in different dataset and research

The features of the dataset play a crucial role in the model building process. If the number of the feature is too much then there is a chance that the model will suffer from overfitting problem. On the other hand in case of very few numbers of feature the accuracy may drop, it may also introduce biases in the result. So, selecting an optimal set of the important attribute for the model building process is very important. Sometimes attribute importance is calculated to rank the features. Table 5 shows a list of some of the features or attributes found in different research.

Table 5. List of attributes

According to [14], we found these unique attributes 1. Customer Id	According to [8], we found some other new attributes: 21. ATM/POS Terminal Number	According to [10], we found some other new attributes: 33. Process code 34. Type of transaction	According to [1] and [2], we found some other new attributes: 50. Credit Limit 51. Education
---	---	---	--

2. Authentication type	22. Account Number	35. Terminal Identifier	52-57. History of last six-month payment
3. Current balance	23. Date of Transaction	36. Merchant Identifier	58-63. Amount of bill for last six month
4. Average bank balance	24. Credit Card Number	37. POS Operation Type	64-69. Amount of payment for last six month.
5. Times of Overdraft	25. ATM Flag	According to [5], we found some other new attributes:	70. Default
6. Credit card age	26. The Total Amount of Transactions of this Card	38. Status of existing checking account	According to [34],
7. Deducted amount	27. The Number of Overdraft Transactions of this card on the Same Day	39. Credit history	71. Repayment Status
8. Location of CC used	28. The Number of Transactions of this card in a Week	40. Saving account or bond	
9. Time of the CC used with respect to the location	29. The Total Amount of Transactions of this Card	41. Installment rate in percentage of disposable income	
10. Average daily Overdraft	30. The Average Transaction Amount of This Card	42. Other debtors or guarantors	
11. Amount of transaction	31. The Number of Overdraft Transactions of this Card in a Week	43. Present residence since	
12. Credit card type		44. Present employment since	
13. The Time of using credit card		44. Other installment plans	
14. Cardholder income		45. Duration in month	
15. Cardholder age		46. Housing	
16. Cardholder position		46. Purpose	
		47. Foreign Worker	
		48. Telephone	

17. Cardholder profession	32. Growth Ratio of Doing Transaction in two Consecutive Weeks	49. Property	
18. Cardholder marital status			
19. Average daily spending			
20. Card frequency			

2.2.9 Performance Evaluation Metrics

There is a common set (e.g., accuracy) of performance evaluation metrics used in different research. In some research, a slightly different performance evaluation metrics are also used. Almost all the performance evaluation metrics are deduced from the confusion matrix. According to [24] [39] [30] [31] [32] and our findings, we found following (Table 6) performance evaluation metrics used in financial fraud mining. This is the upgraded version of similar metrics found in [24].

Table 6. Evaluation metrics

Category	Metric	Equation	Description
Classification	Accuracy or Detection Rate	$(TN + TP) / (TP + FP + FN + TN)$	Most widely used classification performance metric. It is the ratio of correctly classified instance to total instances.
	True Negative Rate or Specificity	$TNR = TN/N$	The ratio of negative instances classified as negative to total negative instances. For example, a laboratory test for identifying patient who doesn't have a disease.
	True Positive Rate or Sensitivity	$TPR = TP/P$	The ratio of positives instances classified as positive to total positive instances. For

			example, a laboratory test for correctly identifying a patient who has a disease.
	Precision/Hit rate	$TP / (TP+FP)$	The ratio of instances correctly classified as positive to total instances classified as positive.
	Recall	$TP / (TP+FN)$	It is the fraction of relevant instances that correctly identified.
	False Positive Rate	FP/N	It is the inverse of TPR which can be calculated as 1- specificity
	F-measure	$2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$	The harmonic mean of precision and recall (sensitivity). Also known as F-score or F.
	$F \beta$		A variation of F-measure which applies a weighting factor β with recall and precision.
	Cost minimization		Minimize misclassification cost of an algorithm.
	Cost	$\text{Cost} = 100 * FN + 10 * (FP + TP)$	
	MCC	$MCC = (TP * TN - FP * FN) / ((TP + FP) * (FN + TN) * (FP + TN) * (TP + FN))^{1/2}$	It is good for determining the quality of binary classification.
Statistical	Z-score	$Z = (x - \mu) / \sigma$	It helps to normalize or standardizing variables.
	Sum of squared error	SSE= summation 1..N (actual	It is a way to measure the deviation or variation from the mean.

		observation - forecasted observation) ²	
Association rule	Support		Group of items that commonly occur together in a problem space [24].
	Confidence		The proportion of samples that match a specific rule against the total that includes the antecedent (support) [24].
	Lift		A correlation measure used to determine whether an association rule is useful to the problem [24].
	Conviction		A measure of the inaccuracy of the rule, or the chance of the antecedent occurring without the consequent [24].
Clustering	Hopkins statistics		A measure of the probability that a variable is randomly distributed within a space, used to determine whether a dataset contains significant clusters [24].
Visual	ROC curve		Receiver operating characteristic curve, a two-dimensional graph that provides an easily interpreted visualization of the success of a binary classification method [24]
	AUC		The area under a ROC curve, given between 0 and 1. Coalesces both the true and false positive rates into a single measurement [24].

2.3 Summary

Research in the financial fraud detection area is heavily constrained mainly by the privacy issue of real financial data and techniques adopted by the corporations. Most of the current research uses private datasets and have a non-disclosure agreement with the data providers. Moreover, the available datasets have an undersampling problem as fraud examples are few in millions of records. So, in order to test their machine learning models, most researchers inject frauds into the dataset. This creates a problem in terms of accuracy as this doesn't adapt to most real industry scenario.

In this section, we have summarized most of the machine learning approaches used in fraud detection research. Unfortunately, there is no standard approach or algorithm to address the issues we are concerned about in this work. Choices are specific to the problem with a lot of trial and error. Sometimes time sensitivity and computation cost sensitivity may change the choice of algorithm, and sometimes multiple algorithms or hybrid approaches work better.

In order to address the issue of mining bad credit accounts, we will break the problem into following parts: a) problem definition b) data preparation c) data exploration d) modeling e) result evaluation, and f) analyze different combinations of the algorithm to measure the efficiency and accuracy.

CHAPTER 3

RESEARCH METHODOLOGY AND DESIGN

Our proposed approach computes the risk factor of an account by combining the risk probability from archived data in a data warehouse (OLAP) with the risk probability of a current transaction (OLTP). The risk probability from the archived data or data warehouse is precomputed and is stored as summarized data. Whereas the risk probability from OLTP is computed in real time as transactions occur and combined with the precomputed risk to determine the overall risk factor. Figure 6 shows a high-level diagram of our proposed approach and Figure 7 shows a flowchart of how processing will occur.

Whenever a new transaction occurs in the OLTP system, it is passed through a *Standard Transaction Testing* process that checks whether the transaction deviates from any of standard rules. If the

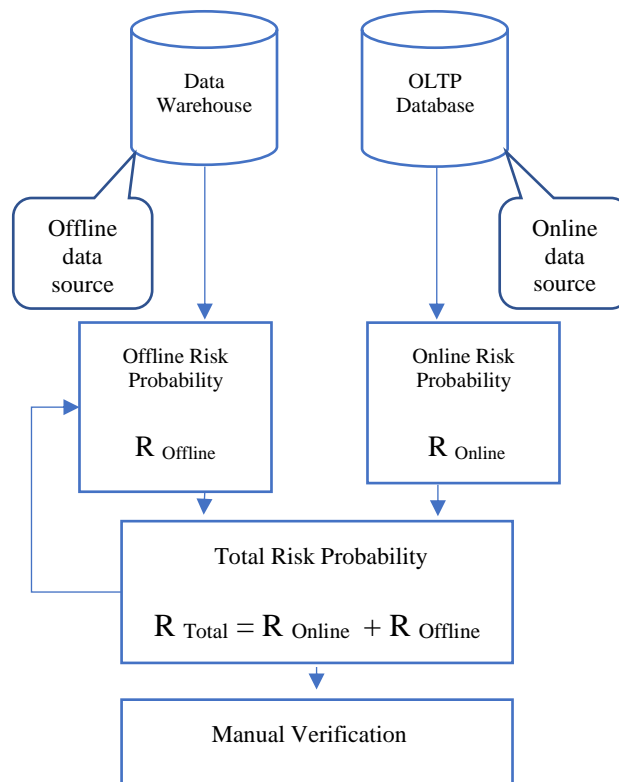


Figure 6. A high-level diagram of the proposed approach

transaction passes the *Standard Transaction Testing*, then no further testing is done and the system continues with the next transaction. However, if the transaction fails the *Standard Transaction Testing*, then the transaction is passed to the *Customer Specific Testing* process where customer specific measures are taken into account to measure the deviation and the risk probability from online data.

Calculating risk probability from the offline data is asynchronous to calculating risk probability from online data. Risk probability from Offline data or OLAP data is a re-calculated one monthly basis to adjust with the customers profile change, like credit limit change, address change, etc. After this recalculation of risk probability, the total risk probability is updated only if the newly calculated total risk

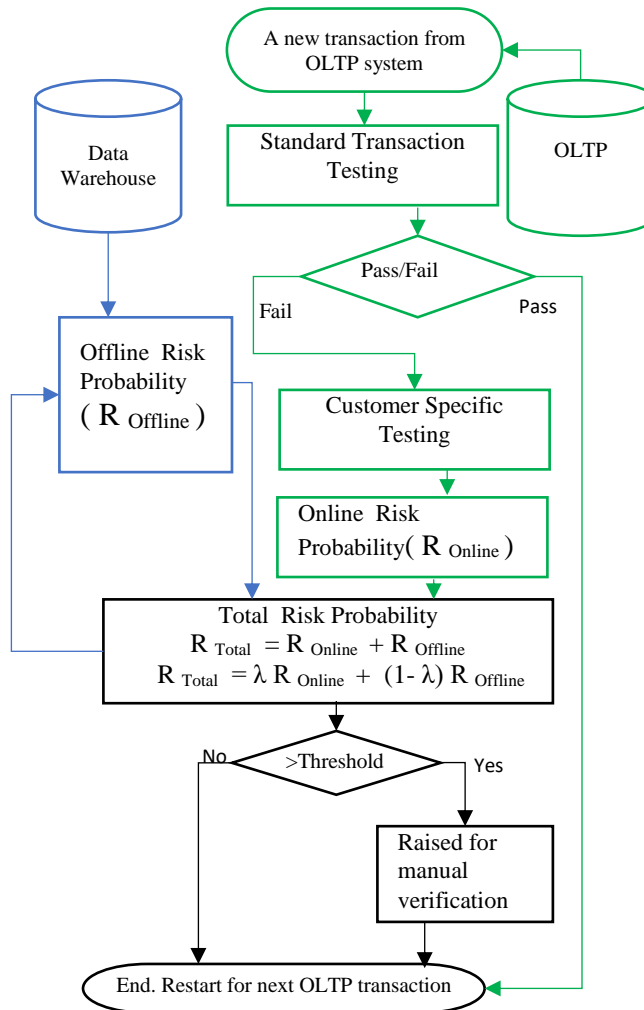


Figure 7. Flowchart of the proposed approach

probability is greater than the stored total risk probability. The online data or OLTP transactions are passed to our approach as a batch of a certain size. The batch size will depend on the capacity of the system. There may be thousands of batch for a particular day. When a transaction occurs in the OLTP system, the combined risk probability is calculated for that transaction which is stored for future use and contributes to the risk probability from offline data for the next transaction from the same account.

At the early stage of our approach, we experimented with various popular classifiers (e.g., Naïve Bayes, J48, Rotation Forest, Extremely Randomized Trees etc.) on the offline data. From our initial experiments, we discovered that *Extremely Randomized Trees (or Extra Trees)* outperform other algorithms in terms accuracy, recall, precision, and F-score. So the result from the best classifier *Extremely Randomized Trees* is used to calculate the risk probability from the offline data. Detailed comparisons of the results from different classifiers are discussed in the result section. We express risk probability from offline data as R_{Offline} and the risk probability form online data as R_{Online} . Risk probability from online data and risk probability from offline data are combined to get the combined risk probability (online + offline). We then express our overall or total risk probability as R_{Total} , which is equivalent to $R_{\text{Online}} + R_{\text{Offline}}$. After subsequent transactions for the same account, the combined risk probability R_{Total} of the current transaction is combined with the offline risk probability for the same account. If the combined risk probability is greater than the threshold then the account is flagged for manual verification, otherwise the process ends here and starts from the beginning for the next transaction. We use the term *combined risk probability* interchangeably with an *overall* or *total risk probability* in rest of this work.

Furthermore, the risk probability from the online data and offline data may carry different weights. For example, giving half of the weight (i.e., 50%) to offline data and the remaining half of the weight (i.e., 50%) to online data might provide better mining results for a particular company or dataset. On the other hand, for another company or dataset, a different combination of offline vs online risk probability weights might be better. So, the modified version of the formula for the total risk probability calculation is:

$$R_{\text{Total}} = \lambda R_{\text{Online}} + (1 - \lambda) R_{\text{Offline}}$$

where λ is the risk factor.

CHAPTER 4

EXPERIMENT SETUP

We experimented with different datasets for this research. This first experiment set (using dataset I & II) shows a proof of concept of our proposed approach. The datasets used for this experiment set are from different sources. The second set of experiments (using dataset III) is done with more realistic data.

4.1 Data

Finding large and interesting sources of financial data is challenging as these data are not made available to the research community because of obvious privacy issues. One of the datasets used in this work is a dataset of a German credit company available publicly on the internet for research purposes [46]. The data contains both a credit summary of 1000 accounts with 24 features or attribute, as well some anonymized detail information. This is a labeled dataset where each account is labeled as good or bad (1 or 0).

Table 7 provides is a description of the attributes under this dataset (Dataset I). In this work, this dataset is also called the offline dataset for experiment set I.

Table 7. Dataset I

Attribute	Type	Example value
Status of existing checking account	Qualitative	No checking accounts, salary assignment for at least 1 year, ≥ 1000
Duration in month	Numerical	12 months
Credit history	Qualitative	No credit taken, all credit paid duly,
Present employment since	Qualitative	<1 year, < 4 years
Personal status	Qualitative	Male: divorced/separated, Female: single/married
Present residence since	Numerical	24 months
Age in years	Numerical	28 years
Housing	Qualitative	Rent, own
Job	Qualitative	Skilled employee, self-employed
Foreign Worker	Qualitative	Yes, no

```

amount, hour1, state1, zip1, field1, domain1, field2, hour2, flag1, total, field3, field4, field
38.85, 8, FL, 342, 4, AFIGECHUD.COM, 1, 8, 0, 38.85, 1688, 8, 0, 0, 0, 0, 1, 0, 1
38.85, 9, GA, 300, 4, FXXRJTGPDTAOLKVEG.COM, 0, 9, 0, 38.85, 1174, 10, 4, 0, 0, 0, 1, 0, 1
12.95, 10, CA, 939, 4, IX.NETCOM.COM, 0, 10, 1, 12.95, -754, 10, 0, 0, 0, 1, 1, 0, 1
12.95, 13, VA, 223, 4, HOTMAIL.COM, 1, 13, 0, 12.95, 3087, 7, 0, 0, 0, 1, 0, 0, 1
12.95, 14, CA, 917, 4, COMCAST.NET, 0, 14, 0, 12.95, 1802, 7, 0, 0, 0, 0, 1, 0, 2
12.95, 14, VA, 201, 4, HOTMAIL.COM, 1, 14, 0, 12.95, -2724, 6, 0, 1, 0, 1, 0, 0, 1
25.9, 14, VA, 201, 4, HOTMAIL.COM, 1, 14, 0, 25.9, -2724, 6, 0, 1, 0, 1, 0, 0, 1
38.85, 15, DC, 200, 4, MAIL.HOUSE.GOV, 1, 15, 0, 38.85, 4743, 8, 0, 1, 0, 0, 0, 1, 2
12.95, 16, FL, 333, 4, AOL.COM, 1, 16, 0, 12.95, -526, 10, 4, 0, 0, 0, 0, 0, 1
12.95, 21, NH, 035, 4, YAHOO.COM, 0, 21, 0, 12.95, -3802, 8, 0, 0, 0, 1, 1, 0, 1
38.85, 13, NJ, 071, 4, SZWJLK.EDU, 1, 13, 0, 38.85, 3478, 8, 0, 0, 0, 1, 0, 0, 1
12.95, 19, GA, 303, 4, AOL.COM, 0, 19, 0, 12.95, 2269, 6, 0, 0, 0, 0, 0, 0, 1
12.95, 22, CA, 935, 4, MINDSPRING.COM, 0, 22, 0, 12.95, -2973, 9, 2, 0, 0, 0, 0, 0, 1
12.95, 9, PA, 190, 4, HPIMZCZTW.COM, 0, 9, 0, 12.95, -8310, 6, 0, 0, 0, 0, 1, 0, 1
38.85, 12, IL, 610, 4, MTQEFPGPYBCG.ORG, 0, 12, 0, 38.85, -881, 7, 0, 1, 0, 0, 0, 0, 1
12.95, 13, WA, 988, 4, CHARTER.NET, 1, 13, 0, 12.95, 3390, 8, 0, 0, 0, 0, 0, 0, 1
12.95, 13, NY, 117, 4, AOL.COM, 0, 13, 0, 12.95, 4835, 8, 2, 0, 0, 1, 0, 0, 1
12.95, 14, PA, 196, 4, AOL.COM, 0, 14, 1, 12.95, -3775, 11, 3, 0, 0, 0, 0, 0, 1

```

Figure 8. Dataset II

To test the computation time of our proposed approach, we need a lot of transactions. For that purpose, we have used a good credit card transaction dataset of 36000 transactions, which is also used in the research [24]. This dataset was used in a data mining contest (UCSD contest 2009) too.

For the set of the experiments, our purpose is to validate our proposed approach. As indicated earlier, that we have been unable to obtain both OLTP and OLAP datasets from the same institution or for the same set of accounts for this research. To tackle this issue, we will implement an approach that decomposes a real credit card default dataset into both OLAP and OLTP datasets by following a real credit card transaction data distribution using the dataset from the UCI machine learning repository which is based on Taiwan’s credit card clients default cases [34]. This dataset has 23 features and 30,000 instances out of which 6,626 (22.1%) are default cases. The features are credit limit, gender, marital status, last 6 months bills, last 6 months payments, and last 6 months re-payment status etc. These are labeled as either 0 (refers to nondefault) or 1 (refers to default). Figures 9 and 10 show a snapshot (5 random records) of the dataset before the decomposition into OLAP and OLTP.

ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	...	BILL_AMT4
18426	18427	150000	2	2	1	24	-1	-1	-1	-1	1596
4614	4615	180000	1	3	1	27	0	0	0	0	5891
16019	16020	360000	2	2	2	31	-2	-2	-2	-2	-7
4282	4283	390000	1	1	1	35	0	0	0	0	49414
16972	16973	450000	1	1	1	67	-2	-2	-2	-2	0

5 rows × 25 columns

Figure 9. Dataset III (Taiwan dataset) part I

BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6	default payment next month
1596	405	0	902	177	1596	405	0	0	0
5891	753	21753	2000	1700	300	0	21000	1000	1
-7	-7	3500	2668	3000	7	0	3507	2500	0
49414	51380	50661	20000	5006	5006	5008	3004	3006	0
0	0	0	0	0	0	0	0	0	1

Figure 10. Dataset III (Taiwan dataset) part II

When we closely look into the dataset, we can see that the payment (PAY_AMT1 to PAY_AMT6) features and bill (BILL_AMT1 to BILL_AMT6) features are actually OLTP transactions, one is of type *payment* and another is type *expenditure*. But the problem is the BILL_AMT is actually the sum of individual transactions for a month. So we decided to break this BILL_AMT into individual transactions by following some real credit card transaction distribution. For example, if BILL_AMT is 2600 dollar for a month, then we want to grab individual transactions from a real credit card transaction dataset that makes the total bill 2600 dollar for a particular month for a particular customer. Here we do have data for six months, starting from April (BILL_AMT6, PAY_AMT6) to September (BILL_AMT1, PAY_AMT1).

We decomposed the dataset into OLAP and OLTP as shown in Figures (11 and 12).

	account	balance_limit	sex	education	marriage	age	total_bill	total_payment	repayment	default
4662	4663	50000	2	3	2	23	28718	1028	0	0
13180	13181	100000	2	3	2	49	17211	2000	0	0
21599	21600	50000	2	2	2	22	28739	800	0	0
1588	1589	450000	2	2	2	36	201	3	-1	0
28730	28731	70000	2	3	1	39	133413	4859	2	0

Figure 11. OLAP dataset created from dataset III

	tid	account	amount	date	type
53664	53665	23665	660	2015-05-29	pay
9327	9328	9328	46963	2015-05-14	exp
37596	37597	7597	3000	2015-05-29	pay
9494	9495	9495	75007	2015-05-14	exp
34112	34113	4113	5216	2015-05-29	pay

Figure 12. OLTP dataset created from dataset III

We created 5 OLTP transactions of type “pay” (payment) from PAY_AMT1 to PAY_AMT5 and 5 OLTP transactions of type “exp” (expenditure) from BILL_AMT1 to BILL_AMT5 from each record of the dataset III. PAY_AMT6 and BILL_AMT6 go into the total_payment and total_bill of OLAP data initially. At the end of the month, the total_payment and total_bill is updated with that month's total bill (BILL_AMT) and total payments (PAY_AMT).

Furthermore, we still need to break down the BILL_AMT attribute into individual transactions by following a realistic transaction distribution. The dataset [45] used in the research [47] is a card transaction dataset from a bank in Spain. We named this dataset as “Spain” dataset. As in Dataset III, we have a total monthly bill, but we do not have individual transactions. So we decided to follow the transaction patterns of this “Spain” dataset [45] to break down the BILL_AMT into individual transactions. Figure 13 is a snapshot of the dataset. If any BILL_AMT of Dataset III matches the monthly total expenditure for a customer in the “Spain” dataset then the individual transactions of that customer are followed to break down the BILL_AMT into individual transactions.

customer	age	gender	zipcodeOri	merchant	zipMerchant	category	amount
'C1425441042'	'2'	'M'	'28007'	'M1888755466'	'28007'	'es_otherservices'	87.67
'C337109624'	'2'	'F'	'28007'	'M1823072687'	'28007'	'es_transportation'	11.17
'C1635613216'	'4'	'F'	'28007'	'M1053599405'	'28007'	'es_health'	105.59
'C996804095'	'3'	'M'	'28007'	'M348934600'	'28007'	'es_transportation'	15.25
'C1331907286'	'2'	'M'	'28007'	'M348934600'	'28007'	'es_transportation'	33.36
'C506520283'	'4'	'F'	'28007'	'M348934600'	'28007'	'es_transportation'	32.96
'C83613815'	'2'	'F'	'28007'	'M348934600'	'28007'	'es_transportation'	16.96
'C1697528836'	'3'	'F'	'28007'	'M1823072687'	'28007'	'es_transportation'	49.91


Figure 13. Spain Dataset

The “Taiwan” and “Spain” Dataset are in different currencies. So we scaled those datasets up/down when needed to convert them into the same scale using the formula below:

$$\text{New value} = (\text{to_max} - \text{to_min}) * (\text{inpt} - \text{from_min}) / (\text{from_max} - \text{from_min}) + \text{to_min}$$

Where, New Value = converted value, to_max = the ceiling of the new range, to_min = the floor of the new range, from_max = the ceiling of the current range, from_min = the floor of the current range, inpt = the value need to be converted.

We faced another issue while preprocessing the data. For some monthly bill amounts in the “Taiwan” dataset, there was no corresponding customer wise sum of the total in “Spain” dataset. So we use *equal frequency binning* to determine the ranges under which a monthly bill amount must fall into. *Equal frequency binning uses* an inverse cumulative distribution function (ICDF) to calculate the upper and lower ranges. Some of the ranges are shown in Figure 14. We took all the unique monthly bill amounts and used that for creating the bins. There were 3888 unique bill amounts which lead to 3888 bins. We are not doing any analysis on the group of customers those has the same total bill amount in a particular month as that is a different direction of research.

Table:  ranges

	index	amount	lower_bound	upper_bound
	Filter	Filter	Filter	Filter
3097	3096	924.22	923.86962962...	924.26679269...
3098	3097	924.45	924.26679269...	924.57191358...
3099	3098	925.05	924.57191358...	925.05405864...
3100	3099	925.07	925.05405864...	925.08013374...
3101	3100	925.12	925.08013374...	925.15643518...
3102	3101	925.3	925.15643518...	925.43949074...

Figure 14. Range determination using equal frequency binning.

For instance, if the monthly total bill (BILL_AMT) for customer X is 923.90 then it falls into the range 923.86962962 – 924.26679269. So, we can follow the transaction distribution for the corresponding customer monthly total expenditure (924.22) in the “Spain” dataset.

4.2 Technology Used

Here is a list of tools that we have used in this work to do the experiment and generate the results:

1. Python 3.5 (Scikit Learn, Pandas): For running Machine Learning algorithms.
2. Python Flask (GUI for visualization)
3. Sqlite3 database (with in-memory option): For storing and manipulation intermediate results and states.
4. D3.js, excel: For generating graphs.
5. Experiments are compatible with both Windows and UNIX based system.

Source code for the implementation of this approach (proof of concept) is available here:

https://github.com/SheikhRabiul/mining_bad_credit_card_accounts

And the source code for the validation of the proposed approach is available here:

https://github.com/SheikhRabiul/mining_bad_credit_card_accounts/tree/master/extension-experiment2

4.3 Experiments

The first step of the experiment is to calculate the offline risk probability R_{offline} from the offline data or OLAP dataset. We also need to select important features from the OLAP data because there may be some features that are redundant or less important to the model. So we need to use some feature ranking techniques. We have used *Random forest Regressor* of *sci-kit learn* for ranking all attributes of the OLAP dataset.

	rank	scaled_rank	feature
0	1.7449	1.000000	purpose
1	0.2394	0.171892	foreign_worker
2	0.1174	0.104785	saving_account_or_bonds
3	0.1130	0.102365	present_residence_since
4	0.0953	0.092629	credit_history
5	0.0523	0.068977	installment_rate
6	0.0518	0.068702	credit_amount
7	0.0166	0.049340	other_debtor_or_guarantors
8	0.0122	0.046920	telephone
9	-0.0012	0.039549	people_for_maintenance
10	-0.0019	0.039164	property
11	-0.0040	0.038009	personal_status_and_sex
12	-0.0091	0.035204	status_of_existing_checking_account
13	-0.0129	0.033113	other_installment_plans
14	-0.0180	0.030308	number_of_existing_credit_this_bank
15	-0.0202	0.029098	housing
16	-0.0284	0.024587	job

Figure 15. Feature Selection

Figure 15 shows the importance of each feature in decreasing order.

We tried different machine learning algorithms on the OLAP data to calculate offline risk probabilities (R_{offline}) from it. Some of the algorithms include Naïve Bayes, J48, Rotation Forest, Extremely Randomized Trees, etc. We selected the most efficient algorithm to calculate the offline risk probability in terms of accuracy, recall and computation time.

Next, we push the OLTP transactions as a batch of a particular size for different set of experiment. The output from OLTP data processing is the online risk probability or R_{online} . For the purpose of getting the risk probability from online data, we present our two methods: *Standard Transaction Testing* and *Customer Specific Testing* (as shown previously in Figure 7). Remember, in order to be a real-time system, each new transaction from the OLTP system is passed through our approach as soon as the transaction occurs.

4.3.1 Standard Transaction Testing

The purpose of this test is to identify transactions that deviate from the normal behavior and pass them to the next test named *Customer Specific Testing*. For the *Standard Transaction Testing*, we have made a *Standard Rule listing* in Table 8. This is a collection of rules that every normal or good transaction requires to follow according to our proposed approach. While this is just an initial set of rules based on the perception, it is possible to add as many as rules needed in this table based on future requirements. *This Standard Rules* table (Table 8) contains rules that reflect standard and normal behavior.

The first rule in Table 8 deals with whether the transaction amount is less than or equal to the summation of the average transaction amount ($\mu_{\text{transaction amount}}$) and the standard deviation of the transaction amount ($\sigma_{\text{transaction amount}}$). The next standard rule regards whether the number of transaction per day for an account is less than or equal to the summation of the average number of transactions per day per account ($\mu_{\text{number of transaction}}$) and the standard deviation of the number of transactions per day per account ($\sigma_{\text{number of transaction}}$). This can help in identifying risky transactions. Other standard rules are included to indicate a common set of rules and are self-explanatory. Each new transaction from the OLTP system is validated

Table 8. Standard Rules

Rule ID	Rule
1	Transaction amount $\leq \Sigma (\mu_{\text{transaction amount}} + \sigma_{\text{transaction amount}})$
2	Number of transaction per day $\leq \Sigma (\mu_{\text{number of transaction}} + \sigma_{\text{number of transaction}})$
3	Payment within due date
4	Minimum amount due paid in last month
5	Paid amount greater than or equal to due amount
6	Transaction location is near user's physical location

according to the standard rules defined in Table 8. The flexibility of our proposed approach allows for users to add as many standard rules as needed. In summary, the *Standard Rule Testing* performs the primary screening of transactions.

Unfortunately, for the first set of experiment, online data or OLTP data is not from same sources or same organization. To express our concept clearly, we have collected some real credit card transactions by anonymizing the identities (hiding account number and other personal information) of the customer. Figure 9 is a snapshot of some real credit card transactions. The features in this dataset are TID (transaction ID), AC (Account), Tran. Date (Transaction Date), Description (Transaction details), Amount (\$) (Transaction amount in the dollar), and category (the category of the transaction).

Table 9. Sample OLTP data

TID	AC	Tran. Date	Description	Amount (\$)	Category
1	1	2017-01-20	SOUTHWES5268506576536 800-435-9792 TX	237.90	Airlines
2	2	2017-01-20	INTERNET PAYMENT - THANK YOU	25.00	Payments and Credits
3	3	2017-01-20	DNH*GODADDY.COM 480- 505-8855 AZDNH*GODADDY.COM	155.88	Merchandise
4	4	2017-01-20	WM SUPERCENTER #657 COOKEVILLE TN	102.88	Supermarket s
5	5	2017-01-20	BESTBUYCOM775203010161 888-BESTBUY MN	131.69	Merchandise

In Table 9, we can see that there is a transaction for account number (AC) 1 with transaction id (TID) 1. And it is a transaction of \$237.90 for an air ticket purchase from Southwest airlines. As soon as the transaction occurs, it is passed to the *Standard Transaction Testing*. All rules of *Standard Transaction Testing* are not applicable to all transactions. There is a relevance mapping table (Table 10) that contains which standard transaction rule is relevant to which type of transaction. Here the type of transaction is determined by the category of the transaction. For the first OLTP transaction, the “Air ticket purchase of \$237.90”, the relevancy mapping and satisfactory result is listed in the table (Table 10). If any rows of the relevancy table (Table 10) have the value “Yes” in the “Relevancy” field for a transaction, it means that the transaction is relevant to the rule. In a similar fashion, if the value of the “Satisfy” field is “Yes”, the transaction satisfies the rule. Now we check to see if rules for which the transaction under test is relevant (Relevancy=Yes) but doesn’t satisfy (Satisfy=No) the rule. That means to search for rows in Table 10, those have the value “Yes” in “Relevancy” column but “No” in the satisfy column. If we can find any such row, then the transaction has failed to pass the *Standard Transaction Testing*. As we can see from the table (Table 10), row 1 and row 4 has the value “Yes” in the “Relevancy” field but “No” in the “Satisfy” field. Thus, in this example, the transaction has failed to pass the *Standard Transaction Testing* and will need to be forwarded to the next test, *Customer Specific Testing*, with a reference that the transaction has failed to satisfy Rule ID 1 and 4 of the Standard Rules table (Table 8).

Table 10. Relevancy Mapping

Rule ID	Rule	Relevancy	Satisfy
1	Transaction amount $\leq \Sigma (\mu_{\text{transaction amount}} + \sigma_{\text{transaction amount}})$	Yes	No
2	Number of transaction per day $\leq \Sigma (\mu_{\text{number of transaction}} + \sigma_{\text{number of transaction}})$	Yes	Yes
3	Payment within due date	No	NA
4	Minimum amount due paid in last month	Yes	No
5	Paid amount greater than or equal to due amount	No	NA
6	Transaction location is near user’s physical location	Yes	Yes

4.3.2 Customer Specific Testing

The *Customer Specific Testing* process is a test that is more customer-centric rather than the standard rules that are applicable to every account in the same way. It takes customer specific measures like foreign national, job change, address change, promotion, salary increase, etc. into consideration. The purpose of this test is to recognize possible causes for which a transaction is unable to satisfy a rule in the Standard Rules. Table 11 represents a listing of some of the possible causes for which a transaction may fail to follow the relevant standard rules in Table 8.

Because not all causes have the same impact. We have created a mapping of customer specific rules with the features in OLAP data. Moreover, each attribute has a coefficient from the feature selection process (Figure 15). If a customer-specific cause is related to multiple attributes of OLAP data, the attribute with maximum among them is chosen.

Returning to our previous example of a transaction of \$237.90 for the air ticket purchase by Account “1”, the transaction fails to pass the *Standard Transaction Testing* due to two reasons: 1) transaction amount was above the summation of average transaction amount and the standard deviation of the transaction amount, and 2) minimum due last month was unpaid. The transaction is then passed to the *Customer Specific Testing* component, along with the offending rules from Table 8 (i.e., Rule ID 1, 4). The *Customer Specific Testing* then checks its customer specific rules table (Table 11) for all rules that contain the value 1 and/or 4 in its “Related Standard Rule” column. From Table 11, we can see that Rule/Cause ID 2 and 4 have the value 1 and/or 4 in their “Related Standard Rule” column, meaning that the rules in row 2

Table 11. Customer Specific Rules

Rule/ Cause ID	Rule/Cause	Related Standard Rule	Impact coefficient
1	Address change	6	.085714
2	Air ticket purchase	1,2	.001905
3	Job switch	3	.083810
4	Out of the country	3,4,1,6	0.100952
5	Foreign Worker	3	0.0869389

and 4 are possible causes of breaking rules 1 and 4 of the Standard Rules (Table 8). So, we have got two possible causes for breaking the rule in *Standard Rules* table (Table 8): 1) air ticket purchase and 2) out of the country. In this case, the customer bought the air ticket but was not out of the country.

Total risk probability for a transaction comes from both online and offline data. So, the equation of total risk probability is as follows:

$$R_{\text{Total}} = R_{\text{Online}} + R_{\text{Offline}} \dots\dots\dots (1)$$

Here,

R_{Total} = Overall risk probability from both online and offline data.

R_{Online} = Risk probability from online data

R_{Offline} = Risk probability from offline data

Brief Result				
	accuracy	precision	recall	fscore
0	0.956	0.956267	0.956	0.955461

Confusion Matrix				
	True Positives	True Negatives	False Positives	False Negatives
0	268	688	10	34

Detail Result			
	account	probability_good	probability_bad
0	1	1.0	0.0
1	2	0.5	0.5
2	3	1.0	0.0
3	4	0.7	0.3
4	5	0.3	0.7
5	6	0.7	0.3

Figure 16. Probability Distribution

We can get the risk probability from offline data ($R_{Offline}$) for corresponding accounts from the value of *probability_bad* shown in Figure 16, which is actually the risk probability distribution value of classification results on OLAP data. For the first transaction of the account $R_{Offline}$ = probability of being bad/default from the probability distribution. Thus, for a transaction N,

$$R_{Offline} = R_{Total \text{ of transaction } N-1}$$

Furthermore, the risk probability from the online data and offline data may carry different weights. For example, giving 60% weight to offline data and 40% weight to online data might provide better mining results for a particular company. On the other hand, for another company, a different combination of offline vs online risk probability weights might be better. So, the modified version of (1) for a total risk probability calculation is:

$$R_{Total} = \lambda R_{Online} + (1 - \lambda) R_{Offline} \dots\dots\dots (2)$$

- Where λ is the risk factor.

For our experiments, we have found that using between 45% and 50% as a weight for the online data, with the remaining % for the offline data weight, provides the best results. In other words, if $\lambda = .45$ or $.5$ then $1 - \lambda = .55$ or $.5$ accordingly. We have used $\lambda = .5$ for our experiments.

To calculate the risk probability from online data (R_{Online}), we have derived the following equation:

$$R_{Online} = [1 - \frac{\sum \text{Impact Coefficient } (X)}{\sum \text{Impact Coefficient } (Y)}] \times 100 \dots\dots\dots (3)$$

X = Relevant *valid* rules from the Customer Specific Rules table (Table 11)

Y = Relevant *valid or invalid* rules from the Customer Specific Rules table (Table 11)

In other words, X is the collection of rules from *Customer Specific Rules* table (Table 11) where the “Related Standard Rule” column has the value of any of the rule ids that are passed from *Standard Transaction Testing* and are valid causes for breaking a standard rule; and Y is the collection of rules from the *Customer Specific Rules* table (Table 11) where the “Related Standard Rule” column has the value of any of the rule ids that are passed from *Standard Transaction Testing* irrespective of whether it is valid

cause or not. If no rule/cause is found in *Customer Specific Rules* table (Table 11) for a transaction that is passed to *Customer Specific Testing*, then the values of X and Y become zero. Thus, the value of R Online from formula (formula 3) becomes 100%, which means the customer has no customer-specific reason in the *Customer Specific Rule* table resulting from assigning the highest online risk probability possible for that transaction. If there were some customer specific reasons, R_{Online} would reduce by some ratio based upon the number of customer-specific causes/rules available and the number of causes/rules among them that are valid for that transaction.

Using the example presented earlier, a customer with id 1 has bought an air ticket but the customer is not out of the country or state yet. Rule id 1 and rule id 4 from standard rule table (Table 8) were relevant to the transaction but not satisfied. That is why the transaction was passed to “*Customer Specific Testing*” with a reference to rule id 1 and 4. In the *Customer Specific Rules*, from Table 11, it is found that the row with “Rule/ Cause ID” 2 and 4 have the value 1 and or 4 in the “Related Standard Rule” column. So, either of the rules *Out of the country* or *Air ticket purchase* from the Customer Specific rules table (Table 11) is the cause of breaking the standard rules 1 and 4 for the transaction we are explaining. That gives us:

Y = { Out of the country, Air ticket purchase }

But the customer’s most recent location, which is usually appended with the OLTP transaction description, says that the customer is not out of the country (yet). So actually, out of the country is not a valid reason for breaking the standard rules, though it is relevant. Thus, with X = { Air ticket purchase }, using the formula (3):

$$\begin{aligned}
 R_{\text{Online}} &= \left[1 - \frac{\sum \text{Impact Coefficient (X)}}{\sum \text{Impact Coefficient (Y)}} \right] \times 100 \\
 &= \left[1 - \frac{\sum \text{Impact Coefficient (Air ticket purchase)}}{\sum \text{Impact Coefficient (Air ticket purchase)+Impact Coefficient (Out of the country)}} \right] \times 100 \\
 &= \left[1 - \frac{1}{1+2} \right] \times 100 \\
 &= .67 \times 100 \\
 &= 67
 \end{aligned}$$

Suppose the offline risk probability that we got for account 1 is 70% (i.e., R_{offline} = 70).

Putting these values into equation (2) and applying risk factors(λ) we get the overall risk probability for account 1 after the transaction 1 is recorded in the OLTP system. Thus, the risk factors(λ) is .7 for our case.

$$\begin{aligned} R_{\text{Total}} &= \lambda R_{\text{Online}} + (1 - \lambda) R_{\text{Offline}} \\ &= .7 \times 67 + .3 \times 70 \\ &= 67.9 \end{aligned}$$

So, for this transaction, there is a 67.9% probability that this account is going to be a bad account.

For this example, we are assuming a *Minimum Total Risk Probability Threshold* of 60% is established beforehand (by the user) based on the analysis of historical data. This means that if the total or overall risk probability is above 60%, then that transaction will be treated as a risky transaction (along with the associated account). In this example, the Overall Risk Probability (R_{Total}) is 67.9% and that is above the threshold 60%, so the account for that air ticket purchase transaction (Transaction 1) is suspended and raised for manual verification to justify the actual nature of the account.

When the overall risk probability for a transaction is completed, the offline risk probability is adjusted based on the value of R_{Total} , which affects the offline risk probability value of the next transaction for the same account. By this way, offline risk probability for an account gradually increases if the customer repeats similar transactions that are passed to the *Customer Specific Testing* from the *Standard Testing*. For both sets of the experiments, the features of the datasets are not exactly same, so there are differences in the number of rules applied in different experiment set. In dataset III, we have all the data (including profile change like credit limit change) from the past six months. So, in the second set of the experiment using dataset III, we updated the risk probability from OLAP data at the end of each month, so that the profile change information comes in to play. In the next chapter, we will show the result for dataset III accordingly (OLTP batch 1, then OLAP batch 1 and then again OLTP batch 2 and so on serially to perceive the result of earlier detection of default accounts). Here each OLTP batch contains transactions for one complete month for better interpretation of the results (earlier detection of default accounts) and better computation time test.

CHAPTER 5

RESULT AND ANALYSIS

In this section, we showed the results that we found from the two different set of the experiments using different datasets. For the first set of the experiment, we showed as much as detail as it shows the working mechanism and results from our proposed approach. And for the second set of the experiment, we showed the important results that are needed to validate our approach.

5.1 Result of experiment set I using Dataset I and II

We have different *Machine Learning* algorithms in order to compare offline risk probabilities. In terms of training and computation time, *Naïve Bayes* outperformed other approaches which are clearly visible from the tables (Table 12 and Table 13) and Figures (Figure 17 and Figure 18). *Random Forest* and *Extremely Randomized Trees* were in second and third position accordingly in terms of total (training and computation) time.

Table 12. Training Time (Dataset I)

Algorithm	Training Time
k-Nearest Neighbor	0.022044182
Support Vector Machine	7.909505844
Random Forest	0.190354347
Naïve Bayes	0.021574259
Gradient Boosting	2.051454782
Extremely Randomized Trees	0.220038652

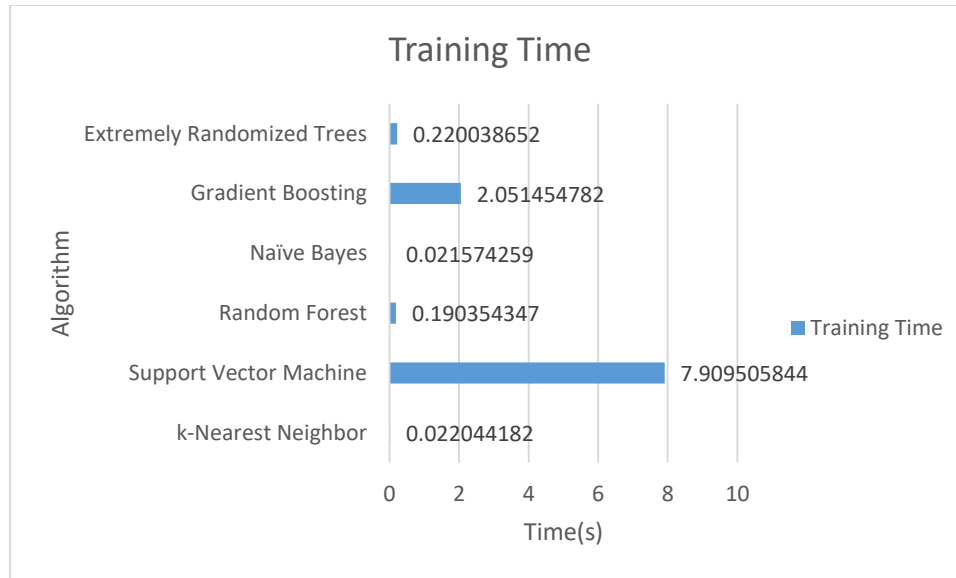


Figure 17. Training Time (Dataset I)

Table 13. Computation Time (Dataset I)

Algorithm	Computation Time
k-Nearest Neighbor	0.10480237
Support Vector Machine	0.03058815
Random Forest	0.011089563
Naïve Bayes	0.002696037
Gradient Boosting	0.003225565
Extremely Randomized Trees	0.012135983

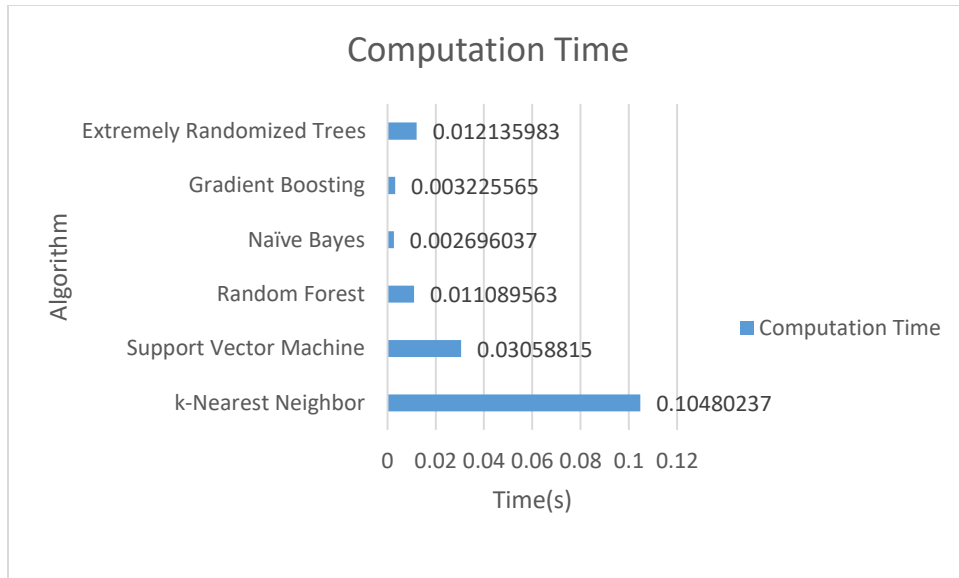


Figure 18. Computation Time (Dataset I)

From Table 14 and Figure 19, we can see that *Extremely Randomized Trees* outperforms other approaches in terms of Accuracy, Precision, Recall, and F-score.

Table 14. Accuracy, Precision, Recall, and F-score

Algorithm	Accuracy	Precision	Recall	F-score
k-Nearest Neighbor	0.806	0.800697	0.806	0.793911
Support Vector Machine	0.776	0.76768	0.776	0.769471
Random Forest	0.936	0.936208	0.936	0.934927
Naïve Bayes	0.75	0.754993	0.75	0.75219
Gradient Boosting	0.865	0.864152	0.865	0.859669
Extremely Randomized Trees	0.954	0.953856	0.954	0.953635

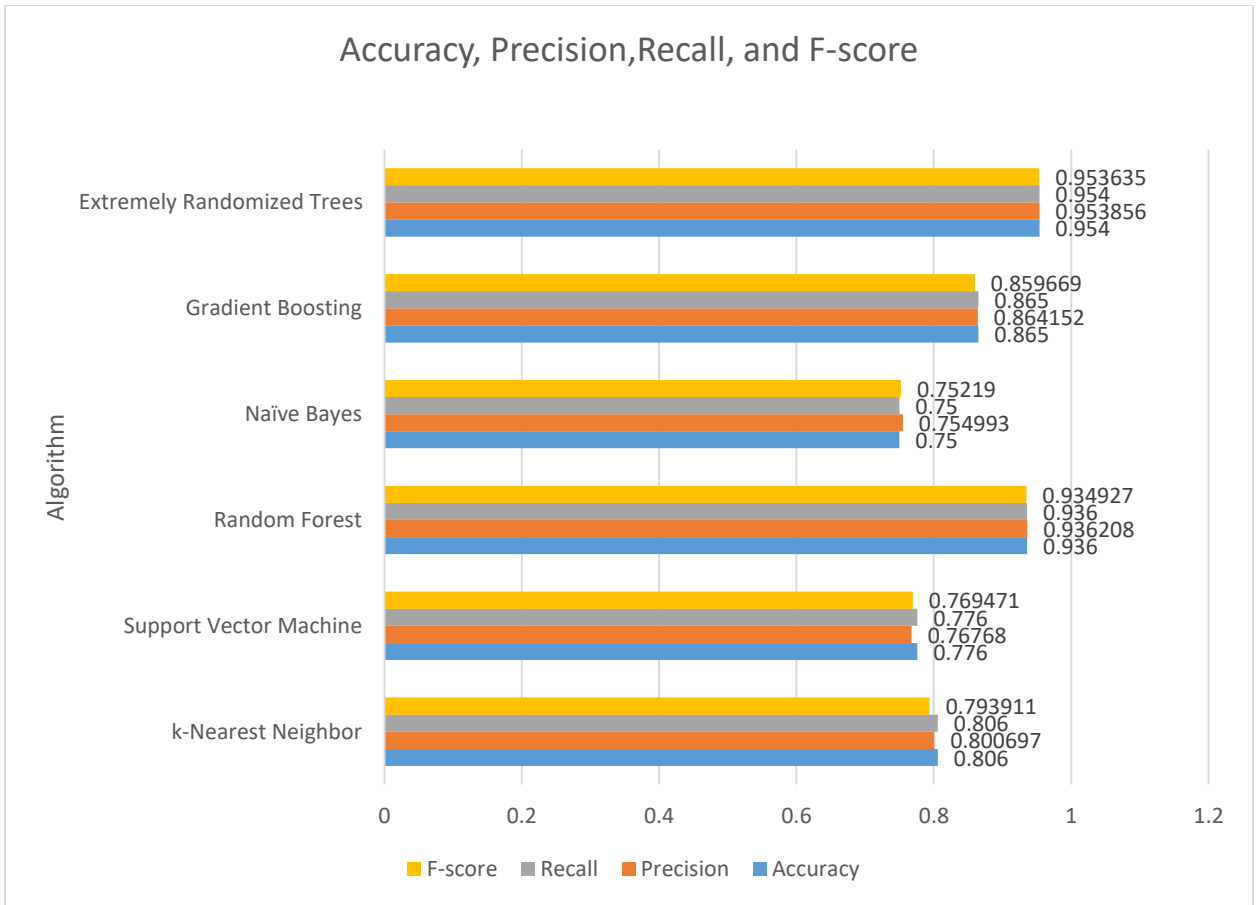


Figure 19. Accuracy, Precision, Recall, and Fscore (Dataset I)

Figure 20 and 21 show the result of *Standard Transaction Testing* and *Customer Specific Testing*. If the total risk probability crosses the risk threshold then that account is flagged as *Bad Account* which needs further verifications.

account	offline	online	event	cause	total	class
6	15.0	31.468531	Paid amount greater than or equal to due amount	Out of the country	46.468531	Good Account
7	0.0	31.118881	Number of transaction per day <= Σ (μ number...	Air ticket purchase , Out of the country	31.118881	Good Account
8	20.0	49.650350	Transaction amount <= Σ (μ transaction amoun...	Air ticket purchase , Air ticket purchase	69.650350	Bad Account
9	0.0	31.468531	Number of transaction per day <= Σ (μ number...	Out of the country	31.468531	Good Account
10	50.0	16.083916	Paid amount greater than or equal to due amount	Job switch , Out of the country	66.083916	Bad Account
11	35.0	49.650350	Number of transaction per day <= Σ (μ number...	Air ticket purchase	84.650350	Bad Account
12	45.0	50.000000	Paid amount greater than or equal to due amount		95.000000	Bad Account
13	25.0	31.468531	Number of transaction per day <= Σ (μ number...	Out of the country	56.468531	Good Account

Figure 20. Flagging accounts for verification

Figure 21 visualizes first few accounts with their status. The red horizontal line is the risk threshold. If the risk probability for any of the accounts crosses the line then it is assumed as a risky account until the flag is cleared. The visualization also helps to realize the intensity of the risk. The higher the bar the higher the risk associated with the account.

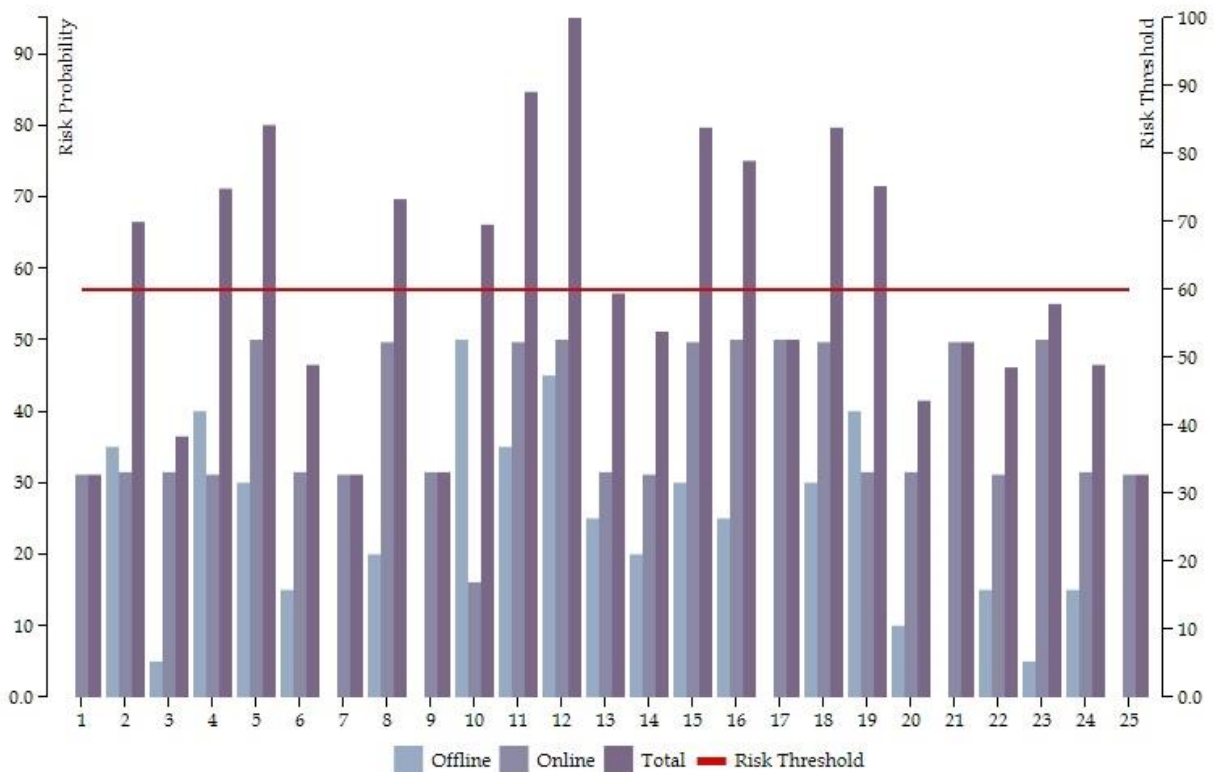


Figure 21. Bad accounts visualization

From the above Figure 21, we can see that some of the accounts are flagged (bars above the red line) as bad accounts using our approach. But we do not have an integrated dataset (a dataset where both OLAP and OLTP comes from the same institution) yet through which we can validate those accounts. To deal with this limitation, we have done a case study on this dataset. We have targeted few accounts (e.g., account 50, 80) and inserted a series of suspicious transactions for those accounts in the OLTP dataset. We try to see whether our approach can identify these accounts as a bad account. Here are some of the suspicious transactions that we mixed with the OLTP dataset. These transactions belong to the different batch of test data as they occurred at different times.

Each of these transactions is suspicious due to one of the reasons below:

1. The transaction is not near user's physical location.
2. Payment amount was less than the total due amount.
3. Transaction amount greater than average+std.

When we run the batches for these transactions in our system, we found that sooner or later all of the accounts for those suspicious transactions were mixed were flagged as a bad account. For instance account 50 is flagged as a bad account after its third attempt of suspicious transaction and account 80 is

Table 15. Suspicious transaction infusion

Account	Amount	Location	Date	Transaction Type
50	38.99	MT	08-01-2017	exp
80	.05	NE	08-02-2017	exp
50	35.00	MT	08-03-2017	pay
80	35.00	TX	08-05-2017	pay
50	5.99	TN	08-06-2017	exp
50	300.88	TN	08-07-2017	exp
80	22.98	TX	08-06-2017	exp
80	11.01	NE	08-10-2017	exp

flagged as a bad account in its 4th attempt of a suspicious transaction. This is one of the important features of using our approach. It can detect fraud attempt which is a combination of multiple less or more suspicious attempt. However, this doesn't validate our approach completely as we are mixing suspicious transactions by ourselves which may lead to possible bias or error or adaptability problem with real fraud scenario. To address this limitation, we experimented with another dataset (Dataset III) in the section ahead.

5.2 Results for experiment set II using Dataset III

As we mentioned earlier (Section 4.1 Data) that we needed to decompose the dataset. Before decomposing the dataset into OLAP and OLTP, we run different algorithms on the whole dataset and we found that the *Extremely Randomized Trees* outperform all the algorithms in terms of Accuracy, Precision, Recall and F-score. The *Extremely Randomized Trees (ET)* algorithm outperformed all previous state of art result [1] [2] on this dataset. The performance gain is mainly due to the fact that the Tree-based approach works very well for some specific types of problem where the number of features is moderate. To the best of our knowledge, this algorithm (*Extremely Randomized Trees*) has not been used on this dataset before. In addition, in terms of execution time *Extremely Randomized Trees* is the second fastest.

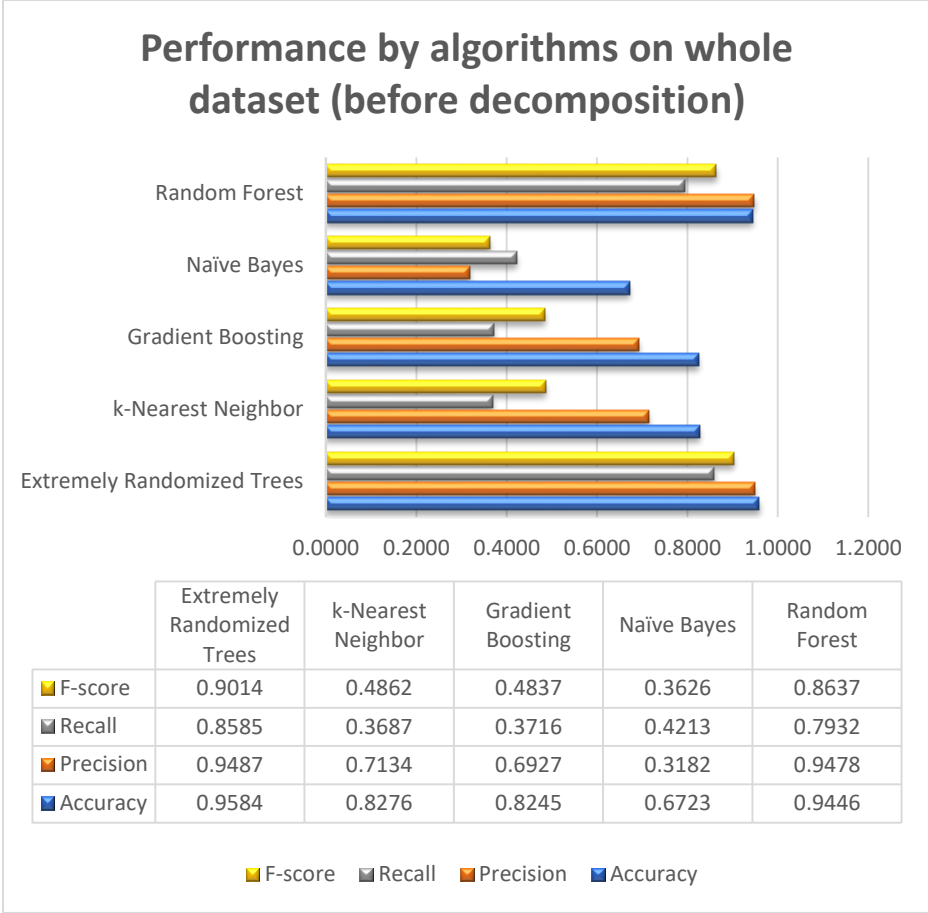


Figure 22. Performance by algorithms on the whole dataset (before decomposition)

We divided the dataset into OLAP and OLTP data, as mentioned in the *Data* section (Section 4.1). We had data bill and payment data for 6 months. Data of the first month was included in the summarized fields (i.e., total_bill and total_payment). From the remaining 5 months data, we made 5 batches of OLAP data and 5 batches for OLTP data. We run OLAP batch 1 and OLTP batch 1 serially. And this way OLAP 2 and OLTP 2 serially next. By this way at the end of batch 5, result from OLTP batch 5 is the final result from the decomposed dataset.

Table 16. Batch wise performance metrics on Dataset III

	Accuracy (OLAP)	Accuracy (OLTP)	Precision (OLAP)	Precision (OLTP)	Recall (OLAP)	Recall (OLTP)	F-score (OLAP)	F-score (OLTP)	Execution Time (OLAP)	Execution Time (OLTP)
Batch 1	0.9450	0.9441	0.9273	0.9189	0.8153	0.8196	0.8677	0.8664	12.2277	152.5444
Batch 2	0.9479	0.9424	0.9318	0.8797	0.8250	0.8565	0.8752	0.8680	9.1385	109.3508
Batch 3	0.9514	0.9396	0.9379	0.8441	0.8356	0.8917	0.8838	0.8672	13.0372	86.8956
Batch 4	0.9518	0.9356	0.9391	0.8205	0.8363	0.9076	0.8847	0.8618	11.5882	89.7383
Batch 5	0.9526	0.9314	0.9361	0.7990	0.8433	0.9215	0.8873	0.8559	10.2138	133.5813

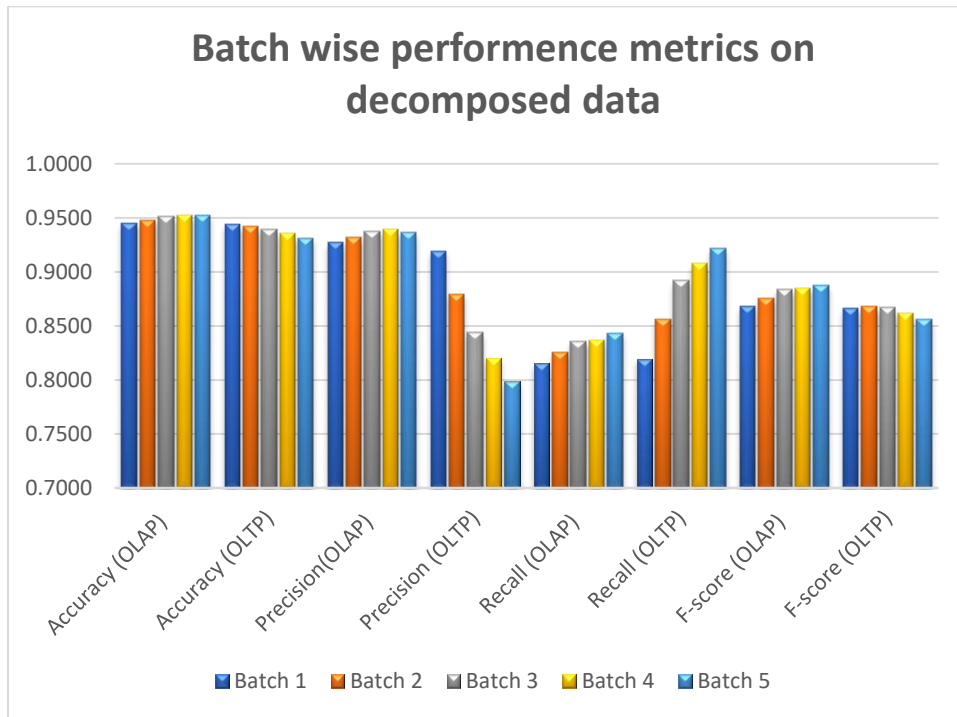


Figure 23. Batch wise performance metrics on Dataset III

From Table 16 and Figure 23 we can see that *recall* for both OLAP and OLTP increases with the number of batches number. This implies that the percentage of target (default account or bad account) detection rate has an increasing order with the batch number.

Table 17. Comparison of the result (Direct approach vs Proposed approach)

Approach	TP	TN	FP	FN	Accuracy	Precision	Recall	F-score
Conventional (ET on whole data)	5697	23056	308	939	0.9584	0.9487	0.8585	0.9014
Decomposed (OLAP +OLTP)	6115	21826	1538	521	0.9314	0.7990	0.9215	0.8559

Table 17 and the Figure 24 show the comparison of performance using a conventional approach (applying only the best classifier, *Extremely Randomized Trees* on the whole dataset without any other test like Standard Transaction Test or Customer Specific Test), our proposed approach, and state-of-the-art. We want to mention that both of the approaches outperform the state of the art result [1] [2] on this dataset. So far we have seen a maximum accuracy of 84% (82% on training data), and maximum recall of 65.54% among all previous research works on this dataset. While our approach has an accuracy of 93.14% and the direct approach that we applied has an accuracy of 95.84%. We also realize a better *recall* percentage too. In fraud or risk, detection *recall* is very important because we don't want to miss fraud or risks. However, maximizing recall introduces an increase of *False Positives*, which is expected in risk analytics.

Another mentionable contribution of this approach is the early detection. If we notice the recall of all batches batch 1 to 5 from Table 16, we can see that in the first batch (batch 1) 81.96% of risky accounts were detected (recall 81.86%). And from the data of 4 months later, which means from batch 5 we could detect 92.15% of risky accounts (recall 92.15%) which is an improvement of only 10.19%. So this early detection (4 months earlier) of a majority of fraud (81.96% in batch 1) could help organizations to avoid a great amount of loss.

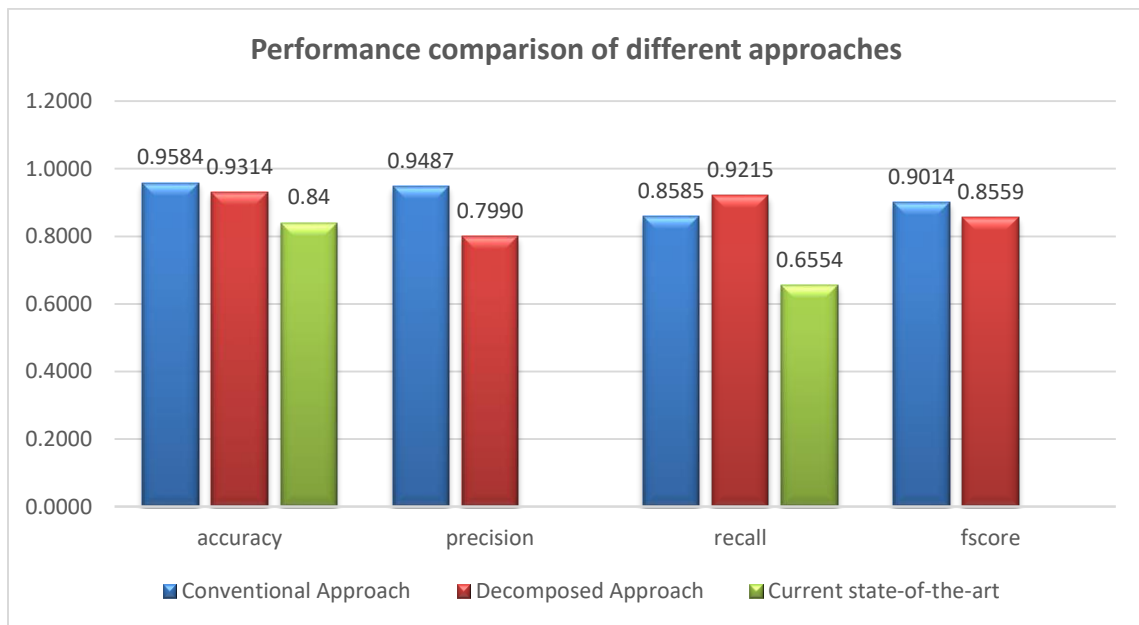


Figure 24. Performance comparison of different approaches

The computation time for both the Conventional Approach and calculating R_{offline} using Extremely Random Trees for 30,000 accounts was on average 11.24 seconds using a commodity laptop with an Intel core i7 processor and 12 GB RAM. Though Naïve Bayes is a bit faster than Extremely Random Trees, its performance in terms of accuracy, precision, recall, and F-score are not. For the online data computation, it took on average of 114.42 seconds for a batch size of on average of 359,583 transactions. For our interpretation of results, we created only one batch per month. However, there is nothing in our proposed approach that requires batches of this size, and any number of transactions per month for online data could be used, which could lead to batches with a much smaller number of transactions with less computation time. To verify this, we tried with batches of different sizes (reducing the batch size by half each time) and we found that the computation time for the online data reduces almost linearly with the reduction of the number of transactions per batch. From Fig. 25, we can see that the trend line (dotted line) is almost in line with the actual line. This demonstrates how fast this approach can process the online transaction and give a decision in near real-time.

Another contribution of this approach is the early detection. While there is ~10% improvement in recall from the first month (batch 1) to the fifth month (batch 5) – from a recall of 81.96% to 92.15% as shown in Table 16 - it is clear that we can achieve a good recall very early in the process, enabling a real-time system to detect potential credit card default.

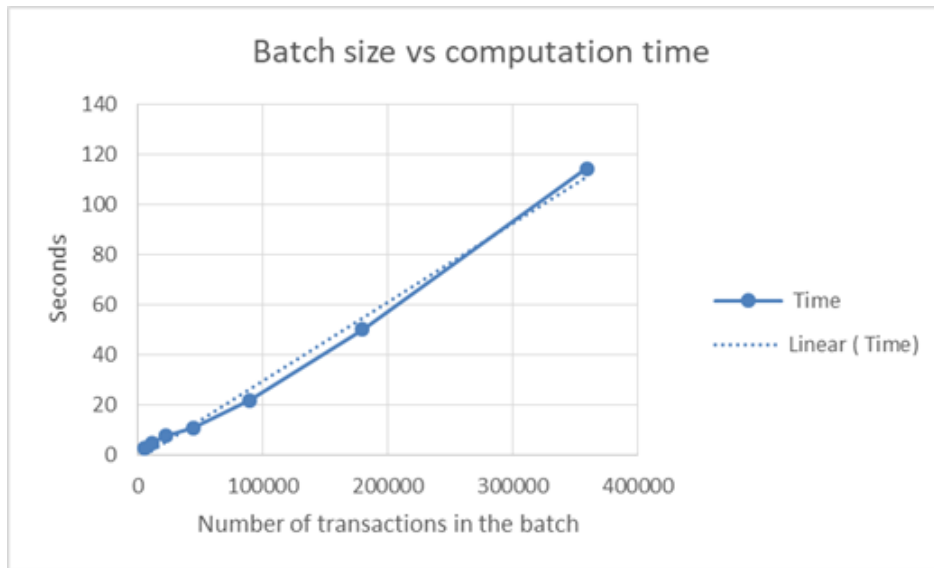


Figure 25. Batch size vs computation time

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this research, we have proposed an approach for mining bad credit card accounts from both OLAP and OLTP data. The main idea of our approach is to calculate the risk factor from the recent transactional data and combine the results with precomputed risk factors from historical data in an efficient way. To make the process efficient, we process a transaction no more than once in the lifetime, once a transaction is processed, it is never considered for future use. Only the combined risk factor is carried forward for future transactions. We showed that our approach can predict a default account far advance, which is very cost efficient for the organization. The performance using our approach, and the direct approach (applying the best classifier, *Extremely Randomized Trees* on the whole dataset) outperformed the state of the art result [1][2] on the dataset III. It was clearly visible that we can get a very optimal outcome by providing an optimal set of data to the selected algorithms and carrying only the calculated risk factor forward for future risk factor calculation.

6.1 Future Work

In our current research, we have defined an approach to mine bad credit card accounts from both OLTP and OLAP data. Some of the improvement and further research that we can do from this point are listed below:

6.1.1 Possible Improvements:

- Multi-level classification (categorize final result as critical, ordinary, under monitoring etc.).
Currently, we are only classifying accounts as good or bad.
- Multi-action (blocking the card, sending SMS notification, calling the cardholder etc.) based on the final risk probability. Currently, we are only freezing the account.
- Focusing more on to reduce the total amount (dollar amount) of fraud instead of just reducing the total fraud count.

6.1.2 Future Research

Some of the future research direction can be as follows:

- A real-time recommendation system to both customer and the company to avoid bankruptcy.
- A real-time fraud detection system.
- We can use credit score and other metrics (e.g., economic situation of the country) which we haven't tried yet to increase the efficiency of the system.
- Incorporating concept drift to deal with the change of new data distribution over time which may affect the effectiveness of the learning model.

REFERENCES

- [1] Yeh, I-Cheng, and Che-hui Lien. "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients." *Expert Systems with Applications* 36.2 (2009): 2473-2480.
- [2] Lu, Hongya, Haifeng Wang, and Sang Won Yoon. "Real Time Credit Card Default Classification Using Adaptive Boosting-Based Online Learning Algorithm." *IIE Annual Conference. Proceedings*. Institute of Industrial and Systems Engineers (IISE), 2017.
- [3] Kirkos E, Spathis C, Manolopoulos Y (2007) Data Mining techniques for the detection of fraudulent financial statements. *Expert Systems with Applications* 32:995–1003. doi: 10.1016/j.eswa.2006.02.016
- [4] A Project Work on e-Banking Fraud. A study on Nigerian Banking sector (2015).
- [5] Liang, D., Tsai, C. F., Dai, A. J., & Eberle, W. (2017). A novel classifier ensemble approach for financial distress prediction. *Knowledge and Information Systems*, 1-26.
- [6] West, J., Bhattacharya, M., & Islam, R. (2014). *Intelligent Financial Fraud Detection Practices: An Investigation*. SecureComm.
- [7] Chen S (2016) Detection of fraudulent financial statements using the hybrid data mining approach. SpringerPlus. doi: 10.1186/s40064-016-1707-6
- [8] Ali Ahmadian Ramaki¹, Reza Asgari² and Reza Ebrahimi Atani (2012). Credit Card Fraud Detection Based on Ontology Graph. *International Journal of Security, Privacy and Trust Management (IJSPTM)*, Vol. 1, No 5.
- [9] G.Apparao, Arun Singh, G.S.Rao, B.Lalitha Bhavani, K.Eswar, D.Rajani (2016) Financial Statement Fraud Detection by Data Mining. *Int. J. of Advanced Networking and Applications*.
- [10] M. Vadoodparast, P. A. R. Hamdan, and D. Hafiz, "Fraudulent Electronic Transaction Detection Using Dynamic KDA Model," (IJCSIS) *International Journal of Computer Science and Information Security*, vol. 13, no. 2, Feb. 2015.
- [11] A. N. Pathak, M. Sehgal, and D. Christopher, "A Study on Fraud Detection Based on Data Mining Using Decision Tree," *IJCSI International Journal of Computer Science Issues*, vol. 8, no. 3, May 2011.
- [12] J. W., Yoon, and C. C. Lee, "A data mining approach using transaction patterns for card fraud detection," Jun. 2013. [Online]. Available: arxiv.org/abs/1306.5547.

- [13] T. Xiong, S. Wang, A. Mayers, and E. Monga, "Personal bankruptcy prediction by mining credit card data," *Expert Systems with Applications*, vol. 40, no. 2, pp. 665–676, Feb. 2013.
- [14] K., & D. (2012). Fraud Detection of Credit Card Payment System by Genetic Algorithm. *IJSER*.
- [15] L., Delamaire, & Abdou, H. (2009). Credit card fraud and detection techniques: a review. *Banks and Bank Systems*, 4(2).
- [16] Adnan M. Al-Khatib, "Electronic Payment Fraud Detection Techniques", *World of Computer Science and Information Technology Journal (WCSIT) ISSN: 2221-0741 Vol. 2, No. 4*, 137-141,2012.
- [17] Altman, E. (2001). Bankruptcy, credit risk and high yield junk bonds, Part 1 Predicting financial distress of companies: revisit
- [18] Humpherys SL, Moffitt KC, Burns MB, Burgoon JK, Felix WF (2011) Identification of fraudulent financial statements using linguistic credibility analysis. *Decis Support Syst* 50:585–594.
- [19] Zhou W and Kapoor G (2011) Detecting evolutionary financial statement fraud. *Decision Support Systems* 50, 570-5.
- [20] Altman, E. (1968). Financial ratios, discriminant analysis, and the prediction of corporate bankruptcy. *Journal of Finance*, 23(4), 589–609
- [21] "Machine Learning Vs. Traditional Statistics: Different Philosophies, Different Approaches". *Datasciencecentral.com*. N.p., 2017. Web. 5 June 2017.
- [22] "Machine Learning Vs Statistics". *Kdnuggets.com*. N.p., 2017. Web. 5 June 2017.
- [23] Pun, Joseph (2011) "Improving Credit Card Fraud Detection using Meta-Learning Strategy." MSc thesis.
- [24] West, Jarrod, and Maumita Bhattacharya. "Some Experimental Issues In Financial Fraud Mining". *Procedia Computer Science* 80 (2016): 1734-1744. Web.
- [25] Chaudhary, Yadav, and Maumita Bhattacharya. " A review of Fraud Detection Techniques: Credit Card". *International Journal of Computer Applications (0975 – 8887) (2012): Volume 45–No.1*.

- [26] Krishna Kumar Tripathi¹, Mahesh A. Pavaskar (2012). Survey on Credit Card Fraud Detection Methods. *International Journal of Emerging Technology and Advanced Engineering*, ISSN 2250-2459, Volume 2, Issue 11.
- [27] Mousa Albashrawi (2016). Detecting Financial Fraud Using Data Mining Techniques: A Decade Review from 2004 to 2015. *Journal of Data Science* 14(2016), 553-570.
- [28] "How the Genetic Algorithm Works- Matlab and Simulink". [Online]. Available: <https://www.mathworks.com/help/gads/how-the-genetic-algorithm-works.html?requestedDomain=www.mathworks.com>. Accessed: July. 13, 2017.
- [29] Ricardo Vilalta, Pavel Brazdil, Christophe Giraud-Carrier. *Meta-Learning - Concepts and Techniques*. *Data Mining and Knowledge Discovery Handbook*, pp.717-73, Chapter · January 2010, DOI: 10.1007/978-0-387-09823-4_36.
- [30] Fawcett T (2006) An introduction to ROC analysis. *Pattern recognition letters* 27, 861-74. Guan S-U and Zhu F (2005) An incremental approach to genetic-algorithms-based classification.
- [31] Han J, Kamber M, and Pei J (2011) In *Data mining: concepts and techniques*. Vol. pp. Elsevier,
- [32] Han J, Kamber M, and Pei J (2011) In *Data mining: concepts and techniques*. Vol. pp. Elsevier, Kantardzic M (2011) In *Data mining: concepts, models, methods, and algorithms*. Vol. pp. John Wiley & Sons,
- [33] "11 Types of Credit Cards Frauds | Consumer Protection.com". [Online]. Available: <https://www.consumerprotect.com/11-types-of-credit-card-fraud/>. Accessed: December. 13, 2017.
- [34] "UCI Machine Learning Repository: Default of credit card clients Data Set". [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients#>. Accessed: December. 23, 2017.
- [35] Bhattacharyya S, Jha S, Tharakunnel K, Westland, JC. *Data Mining for Credit Card Fraud: A Comparative Study*. *Decision Support Systems* 2011; 50: 602–613.
- [36] Gadi MFA, Wang X, do Lago AP. *Credit Card Fraud Detection with Artificial Immune System*. In *ICARIS '08 Proceedings of the 7th International Conference on Artificial Immune Systems* 2008; 119-131
- [37] Mahmoudi N, Duman E. *Detecting Credit Card Fraud by Modified Fisher Discriminant Analysis*. *Expert Systems with Applications* 2015; 42: 2510–2516.

- [38] "Online courses - learn anything on your own schedule | Udemy". [Online]. Available: <https://www.udemy.com/>. Accessed: January. 1, 2018.
- [39] Zojaji, Z., Atani, R. E., & Monadjemi, A. H. (2016). A Survey of Credit Card Fraud Detection Techniques: Data and Technique Oriented Perspective. *arXiv preprint arXiv:1611.06439*.
- [40] "Decision Tree Regression". [Online]. Available: http://saedsayad.com/decision_tree_reg.htm/. Accessed: January. 1, 2018.
- [41] "Probability and statistics EBook". [Online]. Available: <http://wiki.stat.ucla.edu/socr/index.php/EBook>. Accessed: January. 1, 2018.
- [42] "Wikimedia Commons: Under License Creative Commons Attribution-Share Alike 4.0 International ". [Online]. Available: https://commons.wikimedia.org/wiki/Main_Page. Accessed: January. 1, 2018.
- [43] "Principal Component Analysis ". [Online]. Available: <https://georgemdallas.wordpress.com/2013/10/30/principal-component-analysis-4-dummies-eigenvectors-eigenvalues-and-dimension-reduction/>. Accessed: January. 1, 2018.
- [44] "Hidden Markov Model". [Online]. Available: <https://brilliant.org/wiki/hidden-markov-models/>. Accessed: January. 1, 2018.
- [45] "Synthetic data from a financial payment system | Kaggle". [Online]. Available: <https://www.kaggle.com/ntnu-testimon/banksim1/data>. Accessed: February. 9, 2018.
- [46] "UCI machine learning repository: Data set,". [Online]. Available: [https://archive.ics.uci.edu/ml/datasets/Statlog+\(German+Credit+Data](https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data). Accessed: Feb. 20, 2016.
- [47] Edgar Alonso Lopez-Rojas and Stefan Axelsson. BANKSIM: A bank payments simulator for fraud detection research. 26th European Modeling and Simulation Symposium, EMSS 201420141441521
- [48] "Credit Karma: What is a good credit score?," Credit Karma. [Online]. Available: <https://www.creditkarma.com/faq/what-is-a-good-credit-score>. Accessed: Feb. 20, 2016.
- [49] "Introduction to data warehousing concepts," 2014. [Online]. Available: <https://docs.oracle.com/database/121/DWHSG/concept.htm#DWHSG9289>. Accessed: Mar. 28, 2016.

VITA

Sheikh Rabiul Islam was born in Bagerhat, Bangladesh. He received his Bachelor of Science in Computer Science from Islamic University of Technology, Bangladesh in December 2010. He worked as a software developer for 4 years in the telecommunication industry. He started his Ph.D. in Computer Science at Tennessee Tech University as a direct Ph.D. student, which he is still pursuing. He is also working as a Graduate Teaching Assistant in the Department of Computer Science at Tennessee Tech from where he received a Master of Science in Computer Science in May 2018.