**ABSTRACT**

In recent years credit card fraud has become one of the growing problems. A large financial loss has greatly affected individual person using credit card and also the merchants and banks. Machine learning is considered as one of the most successful technique to identify the fraud. This work reviews different fraud detection techniques using machine learning and compare them using performance measure like accuracy, precision and specificity. The paper also proposes a supervised Random Forest algorithm. With this proposed system the accuracy of detecting fraud in credit card is increased. Further, the proposed system uses learning to rank approach to rank the alert and also effectively addresses the problem of fraud detection.

**TABLE OF CONTENTS**

Title Page

Declaration

Certification

Dedication

Acknowledgement

Abstract

Table of Contents

List of Tables

List of Figures

**CHAPTER ONE: INTRODUCTION**

1.1 Background to the Study 4

1.2 Statement of the Problem 8

1.3 Aim and Objectives of the Study 9

1.4 Significance of the Study 9

1.5 Scope of the Study 10

1.6 Limitation of Problems 11

1.6 Definition of Terms 11

**CHAPTER TWO: LITERATURE REVIEW**

2. 1 Credit Card Fraud 13

2.1.1 Merchant Related Frauds 17

2.1.2 Internet Related Frauds 18

2.2 Credit Card Fraud Detection 21

2.2.1 Challenges of Credit Card Fraud Detection 22

2.2.2 Data Driven Fraud Detection 23

2.3 Machine Learning Algorithms 24

2.3.1 Machine Learning: Where Several Discipline meet 24

2.3.2 Machine Learning approaches 26

2.3.3 Validation and Evaluation 30

2.3.4 Machine Learning Algorithm for Credit Card Fraud Detection Techniques 31

**CHAPTER THREE: MATERIALS AND METHODS**

3.1Methodology

3.1.1 Justification of Methodology

3.2 Data Collection

3.2.1 Dataset

3.2.2 Sample of Credit/Debit Cards

3.3 System Analysis

3.3.1 Analysis of the Proposed Systems

3.3.2 System Design

3.3.3 Detailed Design of the proposed System

3.3.4 Flowchart

3.3.5 Random Forest Algorithms

3.3.6 Algorithm for the Proposed System

3.3.7 Data Flow Diagram

**CHAPTER FOUR: RESULT AND DISCUSSION**

4.1 Implementation

4.1.1 Programming Languages

4.2 Integrated Development Environment

4.2.1 System Requirement

4.2.2 Software Installation

4.3 Python modules (packages)

4.4 Data Exploration

4.5 Data Analysis

4.6 Data Preprocessing

4.7 Data Splitting

4.8 Importing Machine Learning Algorithms

4.9 Evaluation of Machine Learning Algorithms

4.10 Deployment of Machine Learning Model

**CHAPTER FIVE: SUMMARY, CONCLUSION AND RECOMMENDATIONS**

5.1 Summary

5.2 Conclusion

5.3 Recommendation

5.4 Contribution to Knowledge

5.5 Suggestion for further Work

**List of Tables**

**Tables Pages**

2.1 Unlabeled Data Examples

3.1 Dataset structure observation

4.1 Different Programming languages Implementation for Credit /Debit Card Fraud dictation

4.2 Hardware system requirement for Anaconda Installation

4.3 Python packages and their respective functions

4.4 Descriptive Statistics of the dataset

4.5 Comparison of the Four Machine learning Models

4.6 Comparison of Model prediction errors

4.7 Precision Comparison

4.8 Comparison of Model Recall / Specificity

4.9 Comparison of model F1\_score

**List of Figures**

**Figures Pages**

2.1. High risk countries facing credit card fraud threat

2.2 General Classification of Credit Card Fraud

2.3: Several Discipline of Machine Learning

2.4: Machine Learning techniques and their required data

2.5 Network topology number of hidden layers and number of nodes in each layer

3.2 Access Bank Debit Card

3.3 Access Bank Credit Card

3.4 Access Bank Prepaid Card

3.5 The general architecture of the proposed system

3.6 Data Flow Diagram of the proposed system.

4.1 Statistical Structure of the dataset

4.2 Correlation between Variables

4.3 Statistical Analysis of Customers age

4.4 Visual representation of customers age with Histogram

4.5 Comparison Domain transactions

4.6 Average Income of customers (Statistical Analysis)

4.7 Histogram of Average Income of Customers

4.8 Amount against Cvv variables for each transaction

4.9 Pearson’s Correlations between variables of the transaction dataset

4.10 Visual representation of the percentage of fraudulent and valid transaction in the dataset

4.11 Transaction amount against Transaction Type

4.12 Visual Comparison of Errors predictions between

4.13 Comparison of Model Accuracy

4.14 Visual comparison of the model precision

4.15 Visual comparison of Recall/Specificity for each Model

4.16 Visual Comparison of F1\_score Model

4.17 Home page of Graphical User Interface (GUI)

4.18 Customer Welcome page

4.19 Customers Transaction page

4.20 Fraud Transaction detected from a customer transaction

4.21 Administrator Login Page

4.22 Admin Transaction Page

4.23 Customer Transaction from the Admin Page (Tracking transaction at real time)

**CHAPTER ONE**

**INTRODUCTION**

**1.1 Background to the Study**

Online banking has continuously position itself as an unavoidable means of financial transaction in our technology-oriented world. It involves conducting financial transactions over a secure website. At the center of online banking is credit and debit card use, which has become a flexible service-on-the-go medium for financial transaction in our modern-day society; resulting in increased credit/debit card fraud as the case may be Nwogu [1].

Despite the huge benefits of deploying and using online banking system, the system has been plagued by enormous vulnerability which has been mainly linked directly to credit/debit card use. Banks, merchants and card issuers are constantly looking for a better improved fraud prevention system for credit and debit card transaction. Due to general belief that information Technology system can go a long way in checkmating and curtailing credit/debit card fraud, banks and merchants have invested lots of resources in fraud prevention research and systems. These show that there has been an increase in research on fraud detection in financial market since 2008 Trivedi [2].

Many credit/debits card fraud detection and prevention techniques have been discussed and implemented in many literatures, with some being deployed successfully. Although these systems have been successful at different times, yet some problems have remained a challenge in development of such systems. This has been identified as the unavailability of real data for fraud prevention research. Other important problems that have been identified are high class imbalance, the availability of few transaction labels by fraud investigators, together with confidentiality issues with the few available data. As a result of this, there has been a renewed interest relating to the development of synthetic data for the training of a fraud prevention system as stated by Bhattacharyya[3].

In the absence of real labeled transaction data, the research community has explored the possibility of replacing real data with synthetically generated data for machine learning research. This unarguably has attracted strong interest in recent times and have supported the usability of synthetic data for the training of various machine learning system, arguing that there is no difference between a synthetic generated data and real data. They equally listed the benefits of using synthetic data. Consequently, there is a belief that synthetic data can solve the problem of real data unavailability the research community Khare [4].

According to Wei [5], who in his work proved that there is no significant difference between synthetically generated data and real data; concluding that synthetically generated data can successfully replace real data especially 'in the field of fraud detection and prevention research.

Similarly, the idea of using machine learning to solve financial fraud and crime has become generally acceptable over the years, as it promises more secure and easier real time security implementation for financial transactions. Fraud detection in credit card transaction deals with the process of ascertaining whether new authorized transactions belong to the class of either fraudulent or genuine transactions. Fraud Detection System (FDS) should be cost-effective and efficient. Also, the cost of transaction screening should not be higher than or close to the loss as a result of the fraud. Many technologies and approach have been used to solve the problem of fraud in credit/debit card transaction systems. Emphasis has now shifted to systems that do not only prevent fraud, but also are able to prevent fraud efficiently with minimal possible cost Awoyemi [6].

A system that prevents fraud but has a higher running cost, such that the cost of preventing the fraud is almost equal to the possible loss in capital due to the fraud is technically useless and would not really find use in modern day implementation. In order to cut down on the costs of detecting a fraud, it is imperative to use expert rules and statistical based models (basically Machine Learning) to process the transaction and run a check to distinguish the genuine transactions from the potentially fraudulent transactions Oluwadare [6].

Abdulla [7] in his work cited that some other systems require fraud investigators to review the transactions with high potential score to run a confirmatory test on the samples. First, transactions are filtered off by checking some essential conditions such as available balance and sufficient balance. The predictive model subsequently scores each transaction in accordance with the given parameters. The model scores each transaction high for high-risk transactions or low for very low risk transactions. The transactions with high fraud score are typically fraud transaction suspects that would generate alerts in the system. Investigators subsequently check the transactions with alerts and provide a feedback for each alert.

These feedbacks are in the form of either true positive (fraud. transaction) or false positive (genuine transaction). With the use of Machine learning (ML) techniques which is more advanced than predictive models, we can discover fraudulent patterns and predict transactions that are most likely to be fraudulent efficiently as stated by Patil [8]. This technique basically infers a prediction model on the basis of a set of examples. The model in most cases is a parametric function, and allows predicting the likelihood of fraud in a transaction given a set of features describing the transaction. Through screening all transaction may likely result in a no fraud situation where all possible frauds are detected and transaction terminated; however, this would not be feasible and cost effective as it probably would be costlier than the cost of the fraud itself.

**1.2 Statement of the problem**

Credit/debit card use has become an integral part of our daily lives. More persons have signed up and are using this service. However, the service has had serious security problems; with billions of dollars in losses due to fraud activities on the system. Consequently, the need for research on fraud detection and prevention has become very imperative. Regrettably, the unavailability of genuine and fraud transaction data has impacted heavily and negatively on fraud detection research. Banks and credit/debit card issuing companies appear to be unwilling to release real customer privacy and confidentially issues. A few available data either have high class imbalance (skewed distribution of the transactions towards the genuine class) or are obsolete Zojaji [9].

Again, Machine learning algorithms do not work well with unbalanced or overlapped class distributions. This has necessitated a renewed interest in training fraud prevention systems with synthetic data. Many machine learning algorithms have been presented for fraud detection in credit card which includes Logistic Regression, decision tree, Naive Bayes, Support Vector Machines and K-Nearest Neighbors. This work seeks to develop a machine learning model that will examine whether the transaction was authorized or fraudulent and then compares the results with the above algorithms. The comparison is made using performance measure accuracy, specificity and precision.

**1.3 Aim and Objectives of the Study**

The aim of this research is to develop an improved model for credit and debit card fraud detection.

The specific objectives are to;

1. conduct a survey on multiple credit cards with aim of collecting data on fraud detection
2. identify key patterns or features from the data set
3. design machine learning models for detecting frauds on credit cards
4. implement the proposed system
5. Evaluate and compare the models with existing models using some testing metrics, which include accuracy, sensitivity, specificity, precision, F1 score and the confusion matrix.

**1.4 Significance of the Study**

The process of searching for fraud is lengthy due to the amount of data involved. In most cases auditors

unknowingly get the information they need from the involved employees who deliberately mislead them and waste their time. With an intelligent agent’s fraud detection system in place to check unusual transactions, the workload is distributed among the agents thus a search is faster and block any transaction suspected to be fraudulent. Since different agents communicate and carry out the verifications done manually, they detect a fraud on the fly, before a transaction fraud is concluded. Without an effective system to check against internal attacks, management of these financial institutions rely on auditors both internal and external to trace the fraud, if they know that one has taken place. The problem is that some fraud can go undetected or by the time they figure out that fraud has occurred it is either too late and the fraudsters have disappeared or they have had enough time to cover they’re and the trail goes cold.

This work is highly significant largely for the following reasons; It will enable us generate and share realistic fraud data within the research community, for fraud detection and prevention research; without breaching confidentially, trust and privacy issues associated with financial transaction data.

The work will introduce a new dimension fraud detection and prevention research with the availability of transaction data, thereby encouraging more research on better systems and techniques for training fraud detection/prevention system.

The work will lead to a more efficient and better optimized system for fraud prevention devoid of the usual class imbalance problem.

Financial institutions can easily set up and implement a fraud prevention system devoid of the intricacies of getting training data for such system.

**1.5 Scope of the Study**

There are a number of financial institutions that grant facilities (credit card) to individuals and corporate bodies, this fact is clearly understood but this work focuses on predictive model using data mining that scores each transaction with high or low risk of fraud and those with high risk generate alerts. Predictive data mining perform interference on the current data in order to make predictions. The intelligent agents check those alerts and provide a feedback for each alert i.e., true positive (genuine) or false positive (fraud). Furthermore, in view of the broad nature of financial fraud, the study is particularly about credit card fraud using data strictly stored on the core banking database and card issuer’s database. This work discussed and implemented a model for improving credit/debit card fraud detection/prevention using a systematic approach. It also involves covering detection, monitoring and response of fraudulent activities in e-commerce business as well as the area of real-time alerting system to enable financial companies stop or deactivate financial transaction suspected to be fraud.

**1.6 Limitations of Problem**

Some of the perceived limitations of the proposed system are as follow:

1. Developing and implementing a credit card fraud detection algorithm poses a challenge of getting necessary dataset from the bank due to clients/customers confidentiality.
2. Some portions of the customer data were not properly recorded which resulted in wide veracity and data inconsistency.
3. Some portion of Missing at Random (MAR), Missing Completely at Random (MCAR), Missing Not at Random (MNAR) where observed in the Access Bank Credit cards details.

**1.7 Definition of Terms**

**CNP**: Cardholder-Not-Present.

**EDA:** Exploratory Data Analysis.

**FDS**: Fraud Detection System.

**FN:** False Negative, when a machine learning model wrongly classifies a positive class as negative in a classification problem.

**FP:** False Positive, when a machine learning model wrongly classifies a negative as positive in a classification problem

**ID3:** Iterative Dichotomiser 3, a decision tree algorithm.

**IID**: Independent and Identically Distributed.

**ML**: Machine Learning.

**OOAD**: Object-Oriented Analysis and Design

**OOD**: Object-Oriented Programming.

**PDF:** Probability Density Function.

**RF**: Random Forest

**SVM**: Supporter Vector Machine, a supervised learning model used for classification and regression.

**TN:** True Negative, when a model classifies a negative class correctly in a classification problem.

**TP**: True Positive, when a model classifies a positive class correctly in a classification problem.

**VoD:** Video on Demand, a service where people pay and download videos online especially movies.

**CHAPTER TWO**

**LITERATURE REVIEW**

**2.1 Credit Card Fraud**

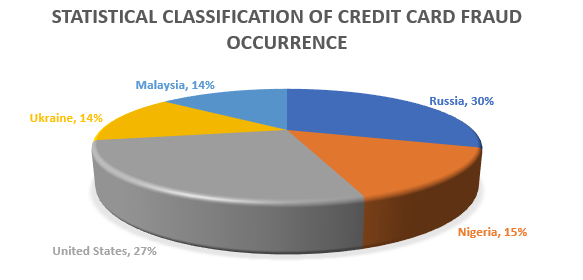
According to Prasad [10], Credit card fraud can be defined as “Unauthorized account activity by a person for which the account was not intended. It is simply a situation where a fraudster uses another person’s credit card for personal reasons while the rightful owner of the card and card issuer remain unaware of the card use. Gregory [11] posits that the annual cost and losses due to credit/debit card frauds run into billions of dollars. Awoyemi [6] wrote that credit card fraud increased enormously between 2005 and 2007, he further claimed that the total credit card fraud in the United Kingdom in the year 2000 stood at *Euro* 286 million, while the United States had a total loss of $3.56 billion in the year 2009, a sharp increase of 10.2% from the previous year’s figure.

Siddhant [12] reports that economic crime and fraud remains an intractable problem for global companies. And according to a report from Association of Certified Fraud Examiners, organizations lose 5% of revenues each year to fraud; this translates to a projected global fraud loss of around $3.7 trillion.

The need for a more robust credit/debit card fraud detection/prevention system has become a great concern to card issuing companies and users.

According to Shirgave [13] fraud costs many firms millions of pounds in losses and fines annually, and has continually been on the increase. Almost all major investment banks have been affected by fraudulent and fraud related credit/debit card transactions over the years. In addition to this, fraudulent activities may impact a business or merchant badly by damaging their reputation, causing a huge drop in the number of customers patronizing their services. HSN Consultants [14] in their report stated that losses to card issuers worldwide reached a whopping $15.72 billion in the year 2015. This was 72% of the total credit/debit card fraud in 2015, which was put at $21.84 billion for the year 2015. This means card issuers bore 72% of the total fraud payback for the year 2015, while the merchants and the customers bore the remaining 28%.

Based on statistical data stated Pozzolo [15], the high-risk countries facing credit card fraud threat is illustrated in Fig 2.1. Russia has the most fraud rate with staggering 30%, which is closely followed by United States at 27% fraud rate. After these two, Nigeria with the rate of 15% is the riskiest country in West Africa and finally are Ukraine and Malaysia which are both 14%. demonstrated in figure 2.1.



**Figure 2.1. High risk countries facing credit card fraud threat Pozzolo [15]**

Incidentally and regrettably, the development of new technologies has provided additional ways and media through which criminals commit fraud successfully. This has become a major problem for financial institutions globally Barse[16] .

Credit/debit card fraud types have been categorized by different authors. Hussein [17] listed the ways in which credit/debit card frauds occur to include the following;

I. Stolen card fraud: this is apparently the commonest type of fraud. Here, the fraudster would try to spend as much as possible and as quickly as possible. Detecting this type of fraud typically relies on the discovery of a deviation from the usual or normal pattern of credit card.

II. Cardholder-Not-Present (CNP): in this fraud type, transactions are performed online or on the internet without the card or the cardholder being present at the point-of-sale. The card owner is usually unaware their card is being used; while the merchant or retailer is unable to physically confirm the card details or the identity of the cardholder. This makes the user unknown and able to disguise their true identity. The details of the credit card are harvested surreptitiously without the cardholder’s knowledge. Such frauds equally demand quick detection to forestall further unauthorized use of the card information and funds.

III. Application fraud: this basically is a scenario where an individual applies for a credit card with false information. This kind of fraud is uncommon as it can be detected easily by simply checking any available records for the correctness of the information supplied by the applicant.

Similarly, Jiaxin [14] has divided credit card fraud into two main types namely;

I. Offline fraud, which they defined as fraud committed by using a stolen physical card at a merchant’s center or any other place with the card holder (in this case the fraudster) being physically present.

II. On-line fraud: defined as fraud committed on the internet, phone or without the card holder being physically present. In this kind of fraud, the fraudster uses the card information to purchase items or pay for services online.

Also Golmohammadi [18] has classified credit card frauds into three categories, namely traditional card related frauds, Internet frauds and merchant related frauds. The different techniques were also listed as follows;

**2.1.1 Merchant Related Frauds**

These frauds are initiated by the merchant or their employees. The following fall into this type of frauds;

I. Merchant Collusion: occurs when the merchant or their employees divulge cardholder’s information to fraudsters who commit fraud using the cardholder’s accounts or their personal information.

II. Triangulation: a form of fraud done on a website. Here, products or goods are offered at discount rates with a promise of shipments before payment. The customer place order by applying their credit card information on the website. The fraudsters in turn use these credit card details to order from a legitimate site.

**2.1.2 Internet Related Frauds**

This describes the credit card frauds that can be executed on the internet.

I. Site cloning: here, fraudsters clone the website of their target merchant such that it looks like the original website. Customers intending to access the merchant website are made to mistakenly access the cloned website thinking they are dealing with the original merchant. The cloned sites receive customer credit card details and uses the card details to commit fraud.

II. False merchant sites: Some website operators hide under the disguise of offering a free or cheap service to their customers. These sites request the customer to fill their credit card details to access the service on the website. Consequently, customers’ credit card information is harvested by these people and transferred to credit card criminal syndicate who commit fraud with the information Zojaji [9].

III. Credit card generators: these are the computer programs that generate valid credit card numbers and expiry dates. These generators work by generating lists of credit card account numbers. The software uses the same mathematical algorithm used by card issuers to generate valid card number combinations. The generated information may be used to clone the credit card.

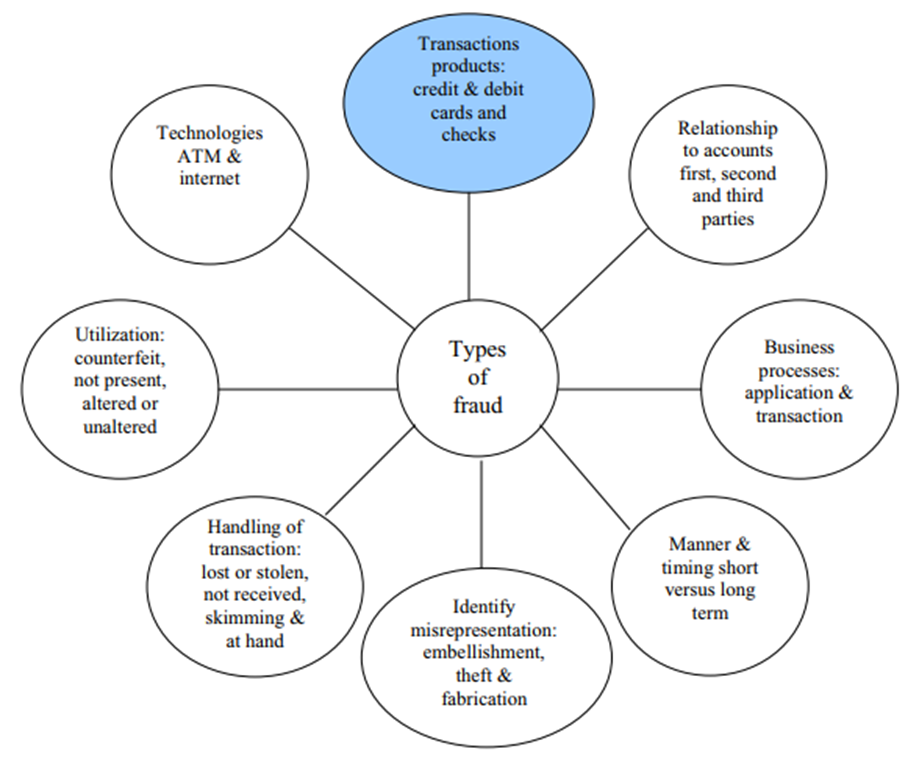
Similarly, Kim [26] also listed other fraud techniques to include the following.

I. Account Takeover: This fraud occurs when a fraudster gets access to a valid customer’s personal information. This they do by either providing the customer’s card or account number. The fraudster subsequently contacts the card issuer, as the genuine cardholder asking for the supplied account email address to be updated with a new email address supplied by the fraudster. The fraudster may also report a card loss and ask for a replacement.

II. Fake and Counterfeit Cards: there also situations where customer’s card is counterfeited using any counterfeiting techniques.

III. Skimming: in skimming, the electronic data on a card’s magnetic stripe is electronically copied onto another. This can be done using many available skimming devices that can copy customer card details when customer card is swiped on the device. The detail is now used to commit fraud.

IV. Phishing: in phishing, customer’s credit card details are harvested. This is usually committed through spam e-mail or pop-up windows appearing to come from the credit card issuer requesting the victim/target to supply some sensitive credit card information. The supplied information is now used to commit fraud. Delaware [19] equally described bankruptcy fraud as a situation where customers use their credit card to spend more than they can actually pay back.



**Figure 2.2 General Classification of Credit Card Fraud Delamaire [19]**

**2.2 Credit Card Fraud Detection**

Fraud detection has largely made use of both expert driven approaches. The Expert Driven approach makes use of domain knowledge from fraud investigators or fraud analysts to define rules that may be used to ascertain whether a new transaction is fraudulent or not. Such systems will typically allow fraud detection experts to use their experience and knowledge garnered over the years on fraud analysis and investigation to formulate rules and conditions that can establish the existence of fraud in a given transaction. The rules are coded into a computer program which in turn acts as the expert. These systems are purely rule based systems, with little or sometimes no intelligence Razooqi [20].

According to Ahmed [21], they make use of the information provided by the expert to solve fraud detection problems in credit card transactions. Assuming from experience, it has been established that the same credit card number cannot be used from two very distant geographical locations over a certain period of time. The expert system will be designed to check whether the credit card use location of the current transaction falls within or very far away from the previous transaction’s geographical location over a specified period of time. The system subsequently assigns a fraud score to the transaction.

Fraud scores are calculated according to the distance of the credit card use location from the previous transaction’s locations. Distant locations receive high fraud scores while close locations receive low fraud score. Transactions with average fraud scores higher than a specified threshold value will be blocked. Fraud investigators will subsequently alert the credit card owners in order to complete investigation of the transaction Patil [8].

Jorgovsky [22] in his work stated that expert driven approaches in credit card fraud detection are used to reduce False Negatives (FN) making sure no frauds are allowed to go undetected. Expert driven systems are simple and can easily be developed once a human expert is willing to supply their knowledge domain for the system’s use. However, these systems have some salient disadvantages in that they are static, and lack the ability to learn new fraud patterns, only detecting known frauds with the rules supplied by the human experts.

**2.2.1 Challenges of Credit Card Fraud Detection**

Fraud detection systems are prune to several difficulties enumerated bellow. An effective fraud detection technique should have abilities to address these difficulties in order to achieve best performance.

I. Imbalanced data: The credit card fraud detection data has imbalanced nature. It means that very small percentages of all credit card transactions are fraudulent. This causes the detection of fraud transactions very difficult and imprecise.

II. Different misclassification importance: in fraud detection task, different misclassification errors have different importance. Misclassification of a normal transaction as fraud is not as harmful as detecting a fraud transaction as normal. Because in the first case the mistake in classification will be identified in further investigations.

III. Overlapping data: many transactions may be considered fraudulent, while actually they are normal (false positive) and reversely, a fraudulent transaction may also seem to be legitimate (false negative). Hence obtaining low rate of false positive and false negative is a key challenge of fraud detection systems.

IV. Lack of adaptability: classification algorithms are usually faced with the problem of detecting new types of normal or fraudulent patterns. The supervised and unsupervised fraud detection systems are inefficient in detecting new patterns of normal and fraud behaviors, respectively.

V. Fraud detection cost: The system should consider both the cost of fraudulent behavior that is detected and the cost of preventing it. For example, no revenue is obtained by stopping a fraudulent transaction of a few dollars.

VI. Lack of standard metrics: there is no standard evaluation criterion for assessing and comparing the results of fraud detection systems Rtayli [23].

**2.2.2 Data Driven Fraud Detection**

Similarly, another way that has been used to implement fraud detection in credit card transaction has been the use of data driven approach. This approach basically implements FDS with machine learning and artificial intelligence techniques. Machine learning techniques would typically allow the computer to learn patterns on their own; thereby making it possible for the computer to recognize fraudulent transaction in a set of credit card transaction data set. Carcillo [24] equally described an artificial intelligence based FDS to be a system that is able to recognize fraudulent transaction after undergoing training.

Also Li [25], Kim [26] equally described data driven systems for credit card fraud detection. These systems are more efficient than traditional expert base systems. They have the ability to learn complex and new fraud patterns on their own and can handle large volumes of dynamic transaction data. Through data driven approaches have been seen to be efficient when it comes to fraud detection in credit card transactions, they still have some salient drawbacks and challenges. These include their inability to handle skewed transactions and the huge data requirement for effective learning to take place.

Another general problem with these systems is the unavailability of transaction data. According Kim [26], to many credit cards issuing organizations are not willing to release their transaction data for FDS development and research. The issue of concept drift is also a problem with data driven approaches which is basically when the system is unable to learn new patterns of fraud as a result of change in the customer behaviors and attack strategies by fraudster.

The system should equally be able to detect changes in customer behaviors and attack strategies. This can only be possible when data is sufficient enough to enable the system learn from it. Many techniques have also fallen into any of the given categories, namely supervised, unsupervised and semi-supervised learning Yousefi [27].

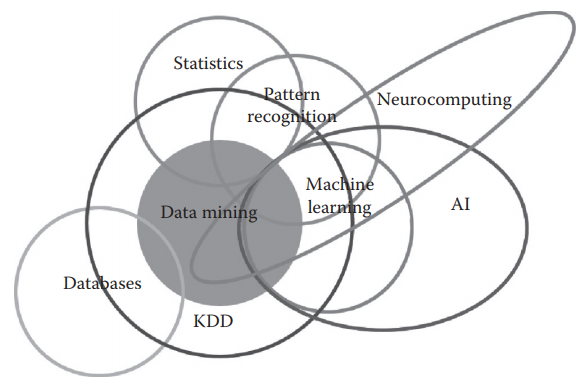
**2.3 Machine Learning Algorithms**

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks Niu [28].

Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step Chandra [29].

**2.3.1 Machine Learning: Where Several Discipline Meet**

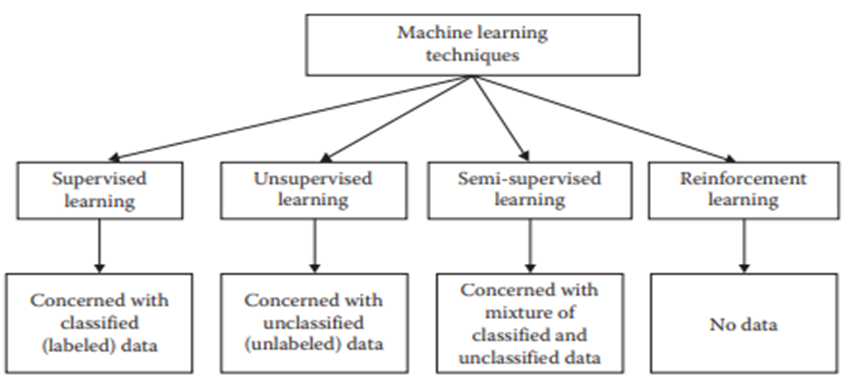
According to Mohammed [30], machine learning is a branch of artificial intelligence that aims at enabling machines to perform their jobs skillfully by using intelligent software. The statistical learning methods constitute the backbone of intelligent software that is used to develop machine intelligence. Because machine learning algorithms require data to learn, the discipline must have connection with the discipline of database. Similarly, there are familiar terms such as Knowledge Discovery from Data (KDD), data mining, and pattern recognition. To view the big picture in which such connection is illustrated below.



**Figure 2.3: Several Discipline of Machine Learning [30]**

**2.3.2 Machine Learning approaches**

Machine learning approaches are traditionally divided into four broad categories, depending on the nature of the "signal" or "feedback" available to the learning system.



**Figure 2.4: Machine Learning techniques and their required data Mohammed [30]**

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E

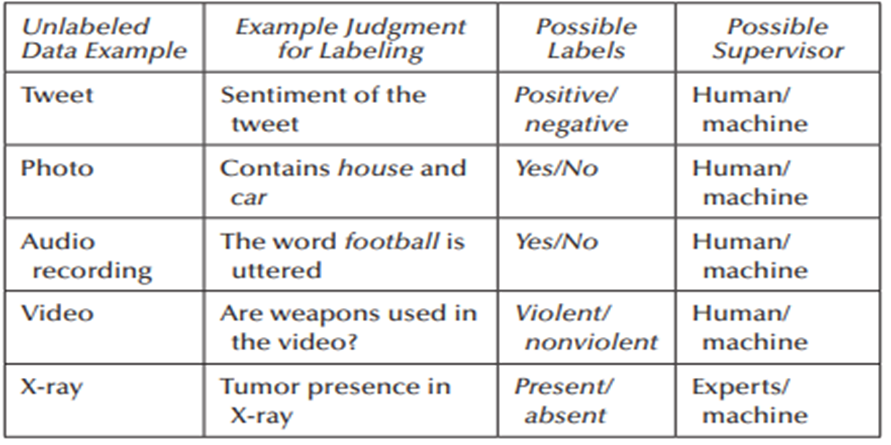
I. Supervised Learning: In supervised learning, the target is to infer a function or mapping from training data that is labeled. The training data consist of input vector X and output vector Y of labels or tags. A label or tag from vector Y is the explanation of its respective input example from input vector X. Together they form a training example. In other words, training data comprises training examples. If the labeling does not exist for input vector X, then X is unlabeled data Thennakoon [31].

Why such learning is called supervised learning? The output vector Y consists of labels for each training example present in the training data. These labels for output vector are provided by the supervisor. Often, these supervisors are humans, but machines can also be used for such labeling. Human judgments are more expensive than machines, but the higher error rates in data labeled by machines suggest superiority of human judgment. The manually labeled data is a precious and reliable resource for supervised learning. However, in some cases, machines can be used for reliable labeling.

Table 2.1 demonstrates five unlabeled data examples that can be labeled based on different criteria. The second column of the table titled, “Example judgment for labeling” expresses possible criterion for each data example. The third column describes possible labels after the application of judgment. The fourth column informs which actor can take the role of the supervisor. In all first four cases described in Table 2.1, machines can be used, but their low accuracy rates make their usage questionable Saia [32].

Sentiment analysis, image recognition, and speech detection technologies have made progress in past three decades but there is still a lot of room for improvement before we can equate them with humans’ performance. In the fifth case of tumor detection, even normal humans cannot label the X-ray data, and expensive experts’ services are required for such labeling. Two groups or categories of algorithms come under the umbrella of supervised learning. They are Regression and Classification Saia [32]

**Table 2.1 Unlabeled Data Example along with Labeling issues Saia[32]**



II. Unsupervised Learning: According Ahmed[21], in unsupervised learning, we lack supervisors or training data. In other words, all what we have is unlabeled data. The idea is to find a hidden structure in this data. There can be a number of reasons for the data not having a label. It can be due to unavailability of funds to pay for manual labeling or the inherent nature of the data itself. With numerous data collection devices, now data is collected at an unprecedented rate. The variety, velocity, and the volume are the dimensions in which Big Data is seen and judged. To get something from this data without the supervisor is important. This is the challenge for today’s machine learning practitioner.

III. Semi- Supervised Learning: In this type of learning, the given data are a mixture of classified and unclassified data. This combination of labeled and unlabeled data is used to generate an appropriate model for the classification of data. In most of the situations, labeled data is scarce and unlabeled data is in abundance (as discussed previously in unsupervised learning description). The target of semi-supervised classification is to learn a model that will predict classes of future test data better than that from the model generated by using the labeled data alone. The way we learn is similar to the process of semi-supervised learning. A child is supplied with:

1. Unlabeled data provided by the environment. The surroundings of a child are full of unlabeled data in the beginning.

2 Labeled data from the supervisor. For example, a father teaches his children about the names (labels) of objects by pointing toward them and uttering their names.

IV. Reinforcement Learning: The reinforcement learning method aims at using observations gathered from the interaction with the environment to take actions that would maximize the reward or minimize the risk. In order to produce intelligent programs (also called agents), reinforcement learning goes through the following steps:

1. Input state is observed by the agent

2. Decision making function is used to make the agent perform an action.

3. After the action is performed, the agent receives reward or reinforcement from the environment.

4. The state-action pair information about the reward is stored.

Using the stored information, policy for particular state in terms of action can be fine-tuned, thus helping in optimal decision making for our agent Mohammed[30] .

**2.3.3 Validation and Evaluation**

Assessing whether the model learnt from machine learning algorithm is good or not, needs both validation and evaluation. If one claims that for a particular training data the function fits perfectly, then for the machine learning community, this claim is not enough. They will immediately ask about the performance of function on testing data. A function fitting perfectly on training data needs to be examined. Sometimes, it is the phenomenon of overfitting that will give best performance on training data, and when yet-unseen labeled data will be used to test them, they will fail miserably. To avoid overfitting, it is common practice to divide the labeled data into two parts:

I. Training data

II. Testing data

A training set is used to build the model and testing set is used to validate the built model. In holdout testing/ validation, one is expected to hold out part of the data for testing. Larger portion of the data is used for model training purpose, and the test metrics of the model are tested on holdout data Kim [33].

The technique of cross-validation is useful when the available training dataset is quite small and one cannot afford to hold out part of the data just for validation purposes. In k-fold cross-validation, the available dataset is divided into k equal folds. Each of these k folds are treated as holdout datasets, and the training of the model is performed on rest of the k − 1 folds. The performance of the model is judged on the basis of holdout fold. The average of performance on all k folds is the overall performance of model as presented by Фесенко [34].

**2.3.4 Machine Learning Algorithms for Credit Card Fraud Detection Techniques**

The credit card fraud detection techniques are classified in two general categories: fraud analysis

(Misuse detection) and user behavior analysis (anomaly detection).

The first group of techniques deals with supervised classification task in transaction level. In these

methods, transactions are labeled as fraudulent or normal based on previous historical data.

This dataset is then used to create classification models which can predict the state (normal or fraud) of

new records. There are numerous model creation methods for a typical two class classification

task such as rule induction, decision trees and neural networks. This approach is proven to

reliably detect most fraud tricks which have been observed before, it also known as misuse

detection Фесенко [34].

The second approach deals with unsupervised methodologies which are based on account behavior.

In this method a transaction is detected fraudulent if it is in contrast with user’s normal behavior.

This is because we don’t expect fraudsters behave the same as the account owner or be aware of the

behavior model of the owner. To this aim, we need to extract the legitimate user behavioral model

(e. user profile) for each account and then detect fraudulent activities according to it Chandra [29].

Comparing new behaviors with this model, different enough activities are distinguished as frauds. The profiles may contain the activity information of the account; such as merchant types, amount, location and time of transactions. This method is also known as anomaly detection. It is important to highlight the key differences between user behavior analysis and fraud analysis approaches.

The fraud analysis method can detect known fraud tricks, with a low false positive rate. These systems extract the signature and model of fraud tricks presented in oracle dataset and can then easily determine exactly which frauds; the system is currently experiencing. If the test data does not contain any fraud signatures, no alarm is raised Ahmed [35].

Thus, the false positive rate can be reduced extremely. However, since learning of a fraud analysis system (i.e., classifier) is based on limited and specific fraud records, it cannot detect novel frauds. As a result, the false negatives rate may be extremely high depending on how ingenious are the fraudsters. User

Behavior analysis, on the other hand, greatly addresses the problem of detecting novel frauds. These methods do not search for specific fraud patterns, but rather compare incoming activities with the constructed model of legitimate user behavior. Any activity that is enough different from the model will be considered as a possible fraud Niu [28].

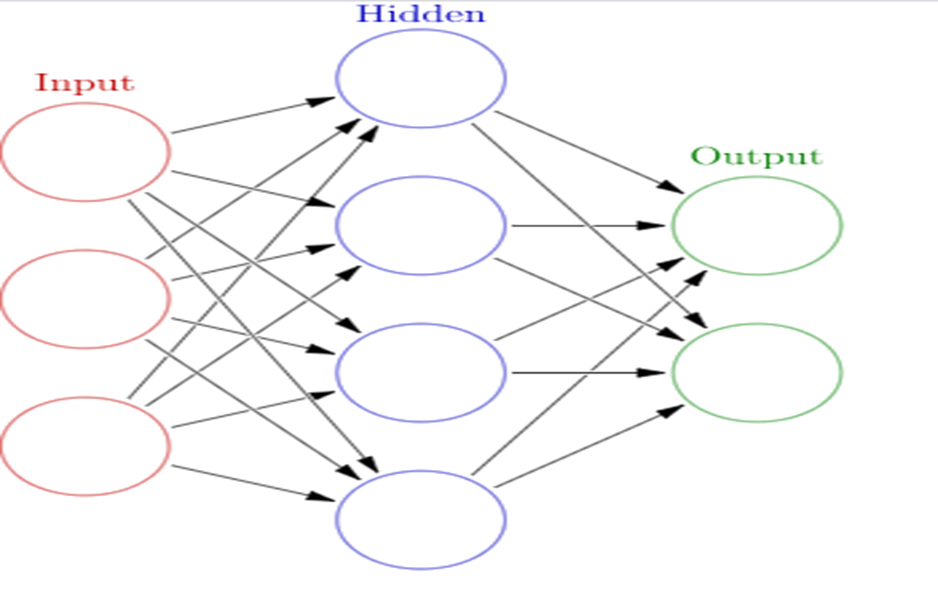
Though, user behavior analysis approaches are powerful in detecting innovative frauds, they really suffer from high rates of false alarm. Moreover, if a fraud occurs during the training phase, this fraudulent behavior will be entered in baseline mode and is assumed to be normal in further analysis Jeragh [33].

In this section we will briefly introduce some current fraud detection techniques which are applied to credit card fraud detection tasks, also main advantage and disadvantage of each approach will be discussed.

I. Artificial Neural Network: An artificial neural network (ANN) is a set of interconnected nodes designed to imitate the functioning of the human brain. Each node has a weighted connection to several other nodes in adjacent layers. Individual nodes take the input received from connected nodes and use the weights together with a simple function to compute output values. Neural networks come in many shapes and architectures. The Neural network architecture, including the number of hidden layers, the number of nodes within a specific hidden layer and their connectivity, most be specified by user based on the

complexity of the problem. ANNs can be configured by supervised, unsupervised or hybrid learning

methods Atiya [35].



**Figure 2.5 Network topology number of hidden layers and number of nodes in each layer Atiya [35]**

**CHAPTER THREE**

**MATERIALS AND METHODS**

**3.1** **Methodology**

The use of effective and appropriate methods in facilitating projects enhances effectiveness and efficiency. The method applied in this research is the Object-Oriented System Analysis and Design method (OOAD) where an existing system is studied from the perspective of objects and similar objects are grouped as classes and their properties are handled as fields while their behaviors are treated as the actions or methods within the same bundle of object. The choice of this methodology is clear since it is a method developed in Software Engineering during the last decades to develop computational models of reality and it is a type of tool needed when one deal with the development of complex computational applications like biological systems. This software development methodology is made up of three aspects, which are Object Oriented Analysis (OOA) that deals with the design requirements and overall architecture of a system which is focused on describing what the system should do as regards to key objects in the problem domain; the second one is the object oriented design (OOD) that translates the system architecture interfaces and classes; and the third one is the object oriented programming (OOP) that implements these programming constructs.

**3.1.1 Justification of Methodology**

First, the use of this methodology helps us to exploit the expressive power of object-based and object-oriented programming languages. Significant improvements in productivity and code quality have consistently been achieved using JAVA, R and Python with a bit of data abstraction thrown in where it is clearly useful. However, further and noticeably larger improvements have been achieved by taking advantage of class hierarchies in the design process. This is often called object-oriented design and this is where the greatest benefits of using Java, Python and R have been found. The experience has been that, without the application of the elements of the object model, the more powerful features of languages such as Smalltalk, C++, Java, Python, R and so forth are either ignored or greatly miss used.

Second, the use of the object model encourages the reuse not only of software but of entire designs, leading to the creation of reusable application frameworks. We have found that object-oriented systems are often smaller than equivalent non-object-oriented implementations. Not only does this mean less code to write and maintain, but greater reuse of software also translates into cost and schedule benefits. However, reuse does not just happen. If reuse is not a primary goal of your project, it is unlikely that it will be achieved. Plus, designing for reuse may cost you more when initially implementing the reusable component. The good news is that the initial cost will be recovered in the subsequent uses of that component.

Third, the use of this methodology produces systems that are built on stable intermediate forms, which are more resilient to change. This also means that such systems can be allowed to evolve over time, rather than be abandoned or completely redesigned in response to the first major change in requirements.

This fourth benefit accrues primarily because integration is spread out across the lifecycle rather than occurring as one major event. The object model’s guidance in designing an intelligent separation of concerns also reduces development risk and increases our confidence in the correctness of our design.

**3.2** **Data Collection**

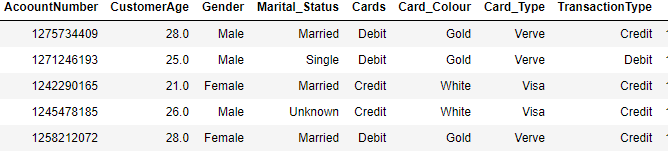
The underlying need for Data collection is to capture quality evidence that seeks to answer all the questions that have been posed. Through data collection business or management can deduce quality information that is a prerequisite for making informed decisions.

Data collection is a structured and systematic process that is compiled on the basis of scientific knowledge to gather information needed especially in answering research objectives. It can also be seen as a methodical process of gathering and analyzing specific information to proffer solutions to relevant questions and evaluate the results. It focuses on finding out all there is to a particular subject matter. This research made used of some samples of Credit/Debit Cards data from Access Bank with strict and entrusted supervision so as to protect customer’s transaction details. The data set is available on request and have undergone proper organization ethics approval processes and available freely for research purposes.

**3.2.1 Dataset**

The Credit/Debit card data set consists of 37,097 observations of 16 attributes. Below is the first five observation of the data set with the first eight variables.

**Table 3.1 shows the first five observations and eight attributes of the data set.**



**3.2.2 Samples of Credit/Debit Cards**

Access bank debit cards: The Access Bank Debit Verve Card allows you make payments directly from your account (only in Nigeria).



**Fig 3.2 Access Bank Debit Card (Source”:** [**https://www.accessbankplc.com/Ways-To-Bank/Cards.aspx**](https://www.accessbankplc.com/Ways-To-Bank/Cards.aspx)**)**

Access bank credit cards: The Access Bank Credit Visa card is a dual-currency denominated payment card which allows you spend and settle in naira for domestic transactions while all your international transactions are billed and settled in US dollars.



**Fig 3.3 Access Bank Credit Card (Source”:** [**https://www.accessbankplc.com/Ways-To-Bank/Cards.aspx**](https://www.accessbankplc.com/Ways-To-Bank/Cards.aspx)**)**

Access bank prepaid cards: The Access Bank Prepaid Card is a multi-currency reloadable payment card used for transactions across multiple channels. The card is available in four currencies; NGN, GBP, USD and Euro, and is accepted globally.



**Fig 3.4 Access Bank Prepaid Card (Source”:** [**https://www.accessbankplc.com/Ways-To-Bank/Cards.aspx**](https://www.accessbankplc.com/Ways-To-Bank/Cards.aspx)***)***

**3.3 System Analysis**

Analysis is an important phase in system development life cycle where factual data are collected in view of understanding the process involved, identifying problems and recommending solutions in order to improve the system functionality. It is an attempt to give birth to new ideas that satisfy the current needs of the user and provide a basis for future improvements. A system is an organized and purposeful collection of components, parts, or elements. These elements are interrelated to make up an integrated whole. Systems have basic components, which are input, processing, and output. The detailed examination of a complex entity and the resulting breaking down of the entity into its constituent parts is termed analysis. The analysis of a system can then be described as a logical reasoning technique that divides the system into its component pieces to evaluate how well these component pieces operate and interact to accomplish their intended purpose. It is a type of system examination which helps an individual understand and make excellently-informed system-related decisions and is also utilized in the production environment. For adequate operational preparation, a form of component examination is often necessary, without appropriate background reviews on the systems used prior to release or deployment, setbacks and shortcomings are bound to arise.

**3.3.1 Analysis of the Proposed Systems**

The proposed system is a complete systematic approach to fraud prevention in credit/debit card systems using Random Forest algorithm. The system is subsequently trained with the data set mention in 3.2.1 and launched in a real credit card transaction network. The system now relies on customer’s fraud feedback reports and fraud investigator analysis to learn new fraud patterns and solve the problem of class imbalance and concept drift in the system. These fraud feedback and reports are important and are used in two ways;

1. To tune the samples for a new insight towards discovering new fraud pattern, thereby solving the class imbalance and the concept drift problem,
2. To tune and update the learning algorithm, thereby solving the problem of concept drift in the learning algorithm which leads to misclassification.

**3.3.2 System Design**

The system design phase is the process of defining the elements that make up the system using pictures or diagrams to portray the functional and non-functional part of the system so as to enable it to be implemented as a proposed working system as desired. The object-oriented approach to system design was used in this work.

**3.3.3 Detailed Design of the Proposed System**

The proposed system design and model has been structured into three subsystems namely;

1. Synthetic data generation subsystem
2. System training and optimization subsystem
3. Classification, feedback/update subsystem

In the synthetic data generation subsystem, a sample of the credit card transaction data is supplied to the system as input. Some parameters (cumulative distribution function, mean of a distribution, variance, covariance and correlation) are extracted from the data set using suitable data analysis tools and software and are used to model and generate synthetic data that will be used in the training of the system. This will be referred to as Real-to- Synthetic-Real approach to data generation in this work.

In the system training and optimization subsystem, the essence of the optimization is to produce optimal and more accurate trees in the forest and to reduce the time needed for the algorithm to converge to a solution.

In the classification and feedback/update subsystem, the system classifies incoming transaction request into fraudulent or non-fraudulent transaction as the case may be. It also receives customers’ feedback and report. The general architecture of the proposed system is given in figure 3.5.

**Data Collection**: The first step in analyzing or building a machine learning algorithm is to import the data set either real operational or transactional data or simulated data.

**Exploratory Analysis**: This refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

**Data Preprocessing**: is a data mining technique which is used to transform the raw data in a useful and efficient format. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.

**Data Splitting**: is the act of partitioning available data into. two portions, usually for cross-validatory purposes. One portion of the data is used to develop a predictive model. and the other to evaluate the model's performance.

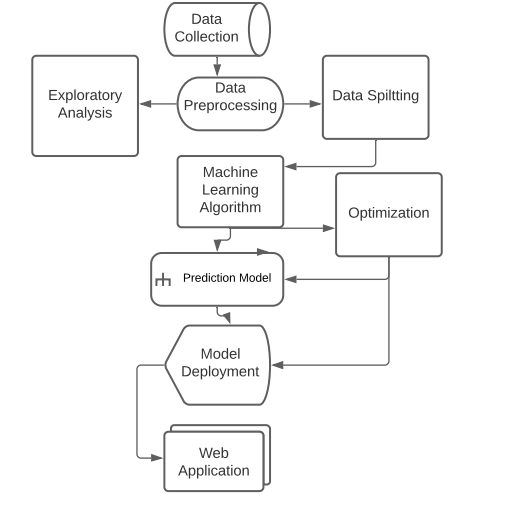
**Building Machine Learning Algorithms**: A machine learning model is built by learning and generalizing from training data, then applying that acquired knowledge to new data it has never seen before to make predictions and fulfill its purpose. Lack of data will prevent you from building the model, and access to data isn't enough. Useful data needs to be clean and in a good shape.

**Prediction Model:** This is the subpart of data analytics that uses data mining and probability to predict results. Each model is built up by the number of predictors that are highly favorable to determine future decisions. Once the data is received for a specific predictor, an analytical model is formulated.

**Model Optimization:** Hyperparameter optimization in machine learning intends to find the hyperparameters of a given machine learning algorithm that deliver the best performance as measured on a validation set. Hyperparameters, in contrast to model parameters, are set by the machine learning engineer before training.

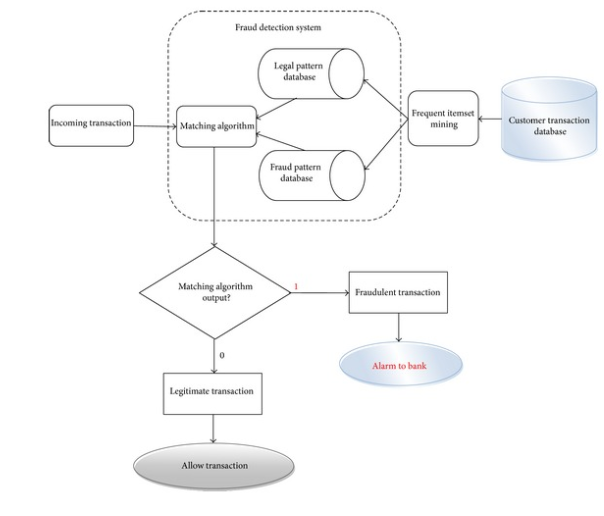
**Model Deployment:** Deployment is the method by which you integrate a machine learning model into an existing production environment to make practical business decisions based on data. It is one of the last stages in the machine learning life cycle and can be one of the most cumbersome.

**Web Application:** A Web application (Web app) is an application program that is stored on a remote server and delivered over the Internet through a browser interface. This research Graphical User Interferes (GUI) was built through the aid of CSS and JAVA script.



**Fig 3.5 The general architecture of the proposed system**

**Flowchart**



**Fig 3.6 Flow chart of the credit/debit card fraud detection**

* + 1. **Random Forest Algorithm**

Random Forest is a tree-based ensemble with each tree depending on a collection of random variables. More formally, for a p-dimensional random vector X = (X1,...,Xp)T representing the real-valued input or predictor variables and a random variable Y representing the real-valued response, we assume an unknown joint distribution PXY(X,Y). The goal is to find a prediction function f(X) for predicting Y. The prediction function is determined by a loss function L(Y, f(X)) and defined to minimize the expected value of the loss

EXY (L(Y, f(X))) (1)

where the subscripts denote expectation with respect to the joint distribution of X and Y. Intuitively, L(Y, f(X)) is a measure of how close f(X) is to Y; it penalizes values of f(X) that are a long way from Y.

Typical choices of L are squared error loss L(Y, f(X)) = (Y − f(X))2 for regression and zero-one loss for classification:

(2)

It turns out that minimizing EXY (L(Y, f(X))) for squared error loss gives the conditional expectation

f(x) = E(Y|X = x) (3)

otherwise known as the regression function. In the classification situation, if the set of possible values of Y is denoted by Y , minimizing EXY (L(Y, f(X))) for zero-one loss gives

(4)

otherwise known as the Bayes rule.

**Algorithm Random Forests**

**Let denote the training data, with xi = (x*i*,1,..., xi.p) T . For j = 1 to J:**

**1. Take a bootstrap sample dj of size *N* from d.**

**2. Using the bootstrap sample d*j* as the training data, fit a tree using binary recursive partitioning (Sect. 2.1):**

**a. Start with all observations in a single node.**

**b. Repeat the following steps recursively for each unsplit node until the stopping criterion is met:**

**i. Select m predictors at random from the p available predictors.**

**ii. Find the best binary split among all binary splits on the m predictors from step i.**

**iii. Split the node into two descendant nodes using the split from step ii.**

**To make a prediction at a new point x,**

**• ˆf(x) = 1 J ∑ j j=1 hˆ j(x) for regression**

**• ˆf(x) = arg max y ∑ jj=1  I(hˆ j(x) = y) for classification where hˆ j(x) is the prediction of the response variable at x using the jth tree .**

**3.3.6 Algorithm for the Proposed System**

**Step** **1**: Get the data from the processed datasets. Check for the datasets is adaptable for the application. Make the data set adaptable for the application.

**Step 2**: Train the data from the data sets which is available for classification. More we train the classification patter better the prediction.

**Step 3:** Test the data, with the better system for training

**Step 4**: Validate the Model

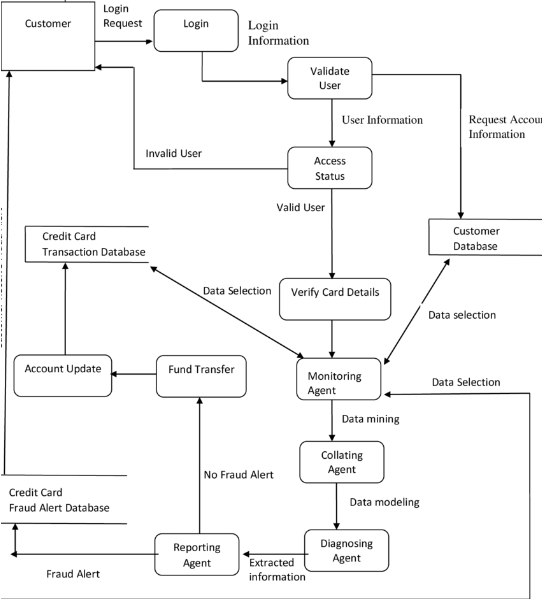
**Step 5**: Optimize the prediction model if need be

**Step** 6: Save the model once

**Step 7**: Deployment of model

**3.3.7 Data Flow Diagram**

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyses an existing system or model a new one.



**Figure 3.9 Data Flow Diagram of the proposed system**

**CHAPTER FOUR**

**RESULTS AND DISCUSSION**

**4.1 Implementation**

The system implementation stage is an important phase of software engineering life cycle. It is the implementation of the system analysis and design done in chapter three. This chapter will involve the programming languages, hardware and software requirements, Integrated Development Environment (IDE), Extraction Transaction and Loading (ETL) processes. It entails building different machine learning algorithms, optimizations, comparisons and deployment of the model to Graphical User Interface.

**4.1.1** **Programming Languages**

A programming language is a formal language comprising a set of instructions that produce various kinds of output. Programming languages are used in computer programming to implement algorithms. Most programming languages consist of instructions for computers. There are two types of programming languages – low-level and high-level. Low-level languages are relatively less advanced and the most understandable languages used by computers to perform different operations. These include assembly language and machine language. While assembly language deals with direct hardware manipulation and performance issues, a machine language is basically binaries read and executed by a computer. An assembler software converts the assembly language into machine code. Low-level programming languages are faster and more memory efficient as compared to their high-level counterparts. The second type of programming languages provides a stronger abstraction of details and programming concepts. Such high-level languages can create code that is independent of the computer type. Moreover, they are portable, closer to human language, and immensely useful for problem-solving instructions.

The table below shows different programming languages that could be used for the implementation of this work. From the languages listed in table 4.1, python was chosen as the choice of language for our implementation due to the fact that it is the most widely used data science programming language in the world today. It is an open-source, easy-to-use language that has been around since the year 1991. This general-purpose and dynamic language is inherently object-oriented. It also supports multiple paradigms, from functional to structured and procedural programming. Therefore, it is [one of the most popular](https://www.upgrad.com/blog/top-6-programming-languages-to-learn/) languages for data science as well. With less than 1000 iterations, it is faster and a better option for data manipulations, machine learning algorithms, deep learning, robotics, Internet of Things etc. Natural data processing and data learning become a cakewalk with the packages contained in Python. Moreover, Python makes it easier for programmers to read the data in a spreadsheet by creating a CSV output.

**Table 4.1 Different Programming languages Implementation for Credit /Debit Card Fraud dictation**

|  |  |
| --- | --- |
| **S/n** | **Types** |
| 1 | Java |
| 2 | Python |
| 3 | R |
| 4 | Scala |
| 5 | C++ |
| 6 | JavaScript |
| 7 | Julia |

**4.2** **Integrated Development Environment**

An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of at least a source code editor, build automation tools and a debugger. Some IDEs, such as NetBeans and Eclipse, contain the necessary compiler, interpreter, or both; others, such as Sharp Develop and Lazarus, do not.

Integrated development environments are designed to maximize programmer productivity by providing tight-knit components with similar user interfaces. IDEs present a single program in which all development is done. This program typically provides many features for authoring, modifying, compiling, deploying and debugging software. For example, code can be continuously parsed while it is being edited, providing instant feedback when syntax errors are introduced, thus allowing developers to debug code much faster and more easily with an IDE. Some IDEs are dedicated to a specific programming language, allowing a feature set that most closely matches the programming paradigms of the language. However, there are many multiple-language IDEs. While most modern IDEs are graphical, text-based IDEs such as Turbo Pascal were in popular use before the availability of windowing systems like Microsoft Windows and the X Window System (X11). They commonly use function keys or hotkeys to execute frequently used commands or macros.

Python is known to work on different kind of editors such as PyCharm, NetBeans, Eclipse, Anaconda etc.

For the purpose of this work, Anaconda will be recommended due to its portability and its usages in multiple different platforms like Windows and Linux operating systems. Anaconda is also known to contain a command line interface and graphical user interface which are known as Jupiter and Spyder respectively.

**4.2.1 System Requirement**

The software requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client’s point of view.

**Table 4.2 Hardware system requirement for Anaconda Installation**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Windows 8 +** | **64-bit macOS 10.13+** | **Linux** |
| System architecture | Windows-64-bit x86, 32-bit x86 | MacOS- 64-bit x86 | Linux- 64-bit x86, 64-bit Power8/Power9 |
| CPU/Memory  (Minimum) | 2 core /4G (RAM) / 100 G (Disk) | 2 core /4G (RAM) / 100 G (Disk) | 2 core /8G (RAM) / 100 G (Disk) |

**4.2.2 Software Installation**

The recommended software for the implementation of this work is Anaconda 2021 on python interpreter version 3.7 and above. Below are the steps for installing the software.

1. Download the Anaconda installer: <https://www.anaconda.com/download/#windows>
2. Double click the installer to launch: To prevent permission errors, do not launch the installer from the Favorites folder. If you encounter issues during installation, temporarily disable your anti-virus software during install, then re-enable it after the installation concludes. If you installed for all users, uninstall Anaconda and re-install it for your user only and try again.
3. Click Next.
4. Read the licensing terms and click “I Agree”
5. Select an install for “Just Me” unless you’re installing for all users (which requires Windows Administrator privileges) and click Next.
6. Select a destination folder to install Anaconda and click the Next button. See FAQ. Install Anaconda to a directory path that does not contain spaces or Unicode characters. Do not install as Administrator unless admin privileges are required.
7. Choose whether to add Anaconda to your PATH environment variable. We recommend not adding Anaconda to the PATH environment variable, since this can interfere with other software. Instead, use Anaconda software by opening Anaconda Navigator or the Anaconda Prompt from the Start Menu.
8. Choose whether to register Anaconda as your default Python. Unless you plan on installing and running multiple versions of Anaconda or multiple versions of Python, accept the default and leave this box checked.
9. Click the Install button. If you want to watch the packages Anaconda is installing, click Show Details.
10. Click the Next button.
11. Optional: To install PyCharm for Anaconda, click on the link to  <https://www.anaconda.com/pycharm> Or to install Anaconda without PyCharm, click the Next button.
12. After a successful installation you will see the “Thanks for installing Anaconda” dialog box:
13. If you wish to read more about Anaconda.org and how to get started with Anaconda, check the boxes “Anaconda Individual Edition Tutorial” and “Learn more about Anaconda”. Click the Finish button.
14. [Verify your installation](https://docs.anaconda.com/anaconda/install/verify-install/).

**4.3 Python modules (packages)**

In programming, a module is a piece of software that has a specific functionality. Modules in Python are simply Python files with a .py extension. Modules are imported from using the import commands as follow:

1. Import [module name]
2. Import [module name] as [aliases]
3. from [module name] import [function]
4. from [module name] import [\*]

The different options can be used to import python modules from its library. Where the module is not defined or found, it can be installed with as follow (Internet required):

1. pip install package name
2. conda install package name
3. pip uninstall package name (to uninstall package)

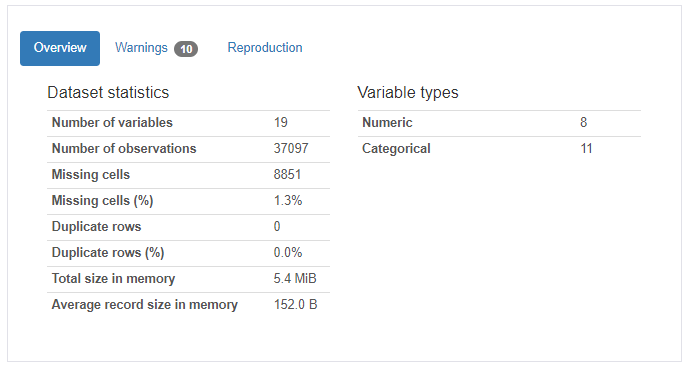
Python is known to have more than ten thousand packages (10,000) which are imported and used for special tasks. In this work, the following well known packages will be used, which are ***Pandas, NumPy, seaborn, matplotlib, SciPy and scikit learn.***

**Table 4.3** **Python packages and their respective functions**

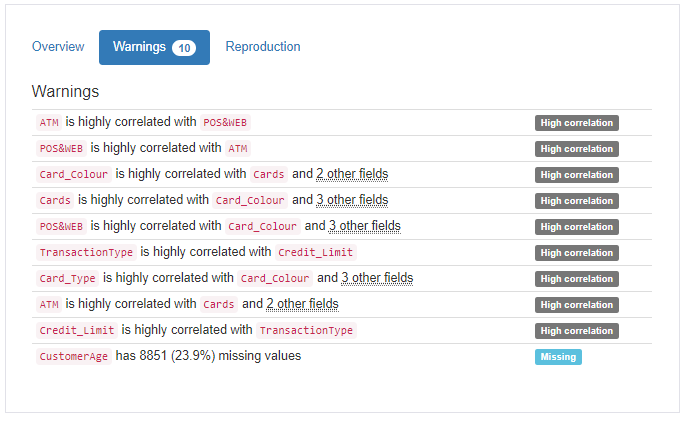
|  |  |  |
| --- | --- | --- |
| s/n | Packages | Functions |
| 1 | Pandas | pandas are open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. |
| 2 | NumPy | a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. |
| 3 | Seaborn | Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. |
| 4 | Matplotlib | Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python |
| 5 | SciPy | SciPy is an open-source Python library which is used to solve scientific and mathematical problems. It is built on the NumPy extension and allows the user to manipulate and visualize data with a wide range of high-level commands |
| 6 | Scikit learn | Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. |

**4.4 Data Exploration**

The approach involves the visual exploration and clear understanding of the characteristics of the data set. This characteristic can include size or amount of data, completeness of the data, correctness of the data, possible relationships amongst data elements or files/tables in the data. From the exploration of our dataset, we had 37,097 observations and 19 variables of which constituted 70,4843 data cells. Out of the 19 variables, the customerAge field contains 8851 null values. The dataset is of 3.5 megabytes (MB) of memory usage.



**Figure 4.1 Statistical Structure of the dataset**

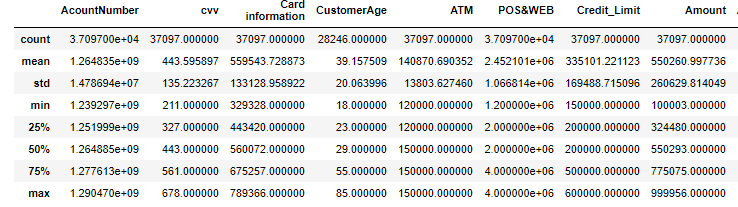


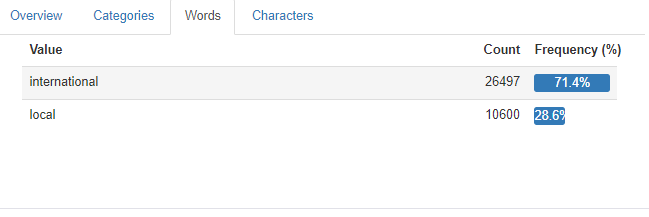
**Figure 4.2 Correlation between Variables**

**4.5 Data Analysis**

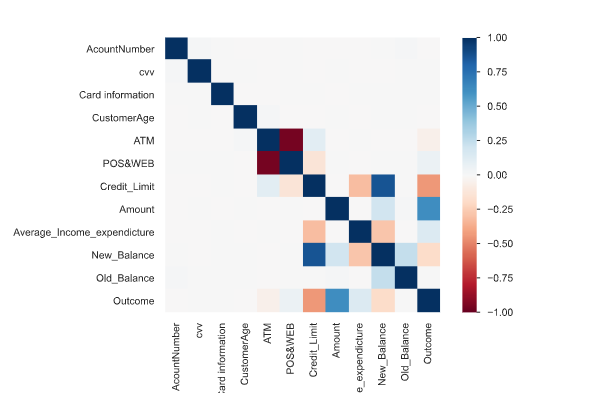
This is the process of inspecting, cleansing, transforming, and modeling data with the goal of discovering useful information, informing conclusions, and supporting decision-making. In statistical applications, data analysis can be divided into descriptive statistics, exploratory data analysis (EDA), and Confirmatory Data Analysis (CDA).

**Table 4.4 Descriptive Statistics of the dataset**





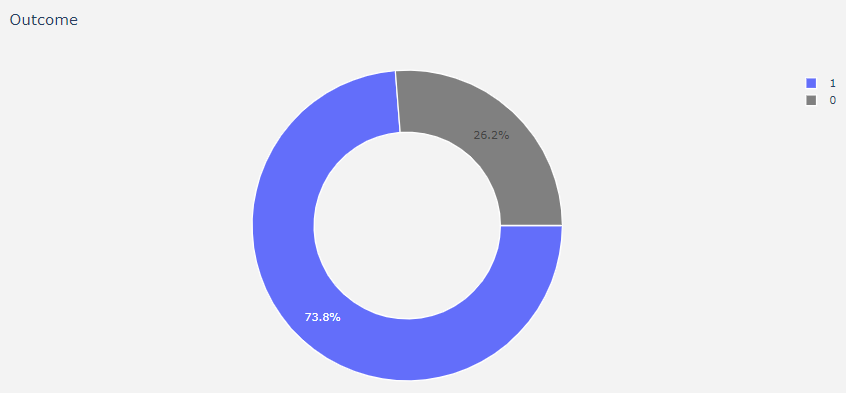
**Figure 4.5 Comparison Domain transactions**



**Figure 4.9 Pearson’s Correlations between variables of the transaction data**

**4.6 Data Preprocessing**

Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format for further analysis and machine learning model building. The dependent variable (Outcome) will be transformed into categorical data. Other non-numerical variables like 'Gender', 'Marital\_Status','Cards','Card\_Colour','Card\_Type','TransactionType', and 'Domain' were also transformed using the pandas ***get\_dummies ()*** function. The CustomerAge column which contains 8,851 missing/ null values were replaced with the mean value of the variable using the ***fillna()*** function in python.



**Figure 4.14 Visual representation of the percentage of fraudulent and valid transaction in the dataset**

**4.7 Data Splitting**

Splitting the data into different sets is technique commonly used in machine learning. The data is usually divided into training and validation set in order to train and find the model hyperparameters (model selection) and estimate the model prediction error or accuracy. The dataset was splitted with the aid of the ***sklearn*** model into training and testing sets respectively. The ratio of the splitting was 80 to 20 for training and testing sets respectively.

**4.8 Importing Machine learning Algorithms**

The machine learning algorithms were imported with the aid of ***sklearn.*** In the course of this research, four machine algorithms were used. The algorithms are; K- Nearest Neighbor, GaussianNB, Logistic Regression and Random Forest. The algorithms mentioned are all classification algorithms. The algorithms were all created, evaluated and compared with some testing metrics, which include accuracy, sensitivity, specificity, precision, F1 score and the confusion matrix.

**4.8.1 Evaluation of Machine learning Algorithms**

A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. The confusion matrix shows the ways in which your classification model is confused when it makes predictions. It gives us insight not only into the errors being made by your classifier but more importantly the types of errors that are being made. It is this breakdown that overcomes the limitation of using classification accuracy alone. In order to test the performance of the proposed system, certain metrics will be used. The following parameters will be used for the testing metrics.

**TP**: True Positive, i.e., the number of positive instances classified correctly by the system

**TN:** True Negative, i.e., the number of negative instances classified correctly by the system

**FP:** True Positive, i.e., the number of negative instances classified correctly by the system

**FN:** True Positive, i.e., the number of positive instances classified correctly by the system

Then,

1. Accuracy measures how accurate the system classifies both classes and is given by

Accuracy, A =  **4.1**

1. Specificity measures how well the system classifies negative samples and is given by

Specificity, SP =  **4.2**

1. Sensitivity/Recall measures how well the system classifies the positive class and is given

Sensitivity/Recall, R=  **4.3**

1. Precision measure the proportion of instances classified as positive class that are indeed

Positive and is given by

Precision, P =  **4.4**

1. F1 Statistics is the defined as the harmonic mean of the sensitivity and precision, and is given

by

F1 = 2 **\* 4.5**

***Note: Where p is Precision and R is Sensitivity/Recall***

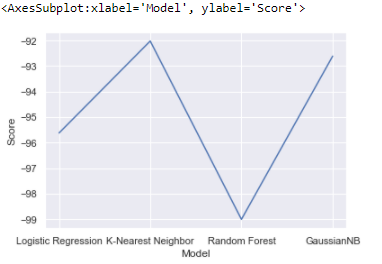
Similarly, to test the accuracy of our models, a couple of statistical test and analysis were be used.

**Table 4.5 Comparison of the Four Machine learning Models**

|  |  |  |
| --- | --- | --- |
| **S/n** | **Model (Accuracy)** | **Scores** |
| 1 | Random Forest | 100.00 |
| 2 | Logistic Regression | 96.62 |
| 3 | GaussianNB | 93.61 |
| 4 | K-Nearest Neighbor | 93.02 |

**Table 4.6 Comparison of Model prediction errors**

|  |  |  |
| --- | --- | --- |
| **S/n** | **Model Error** | **Scores** |
| 1 | Random Forest | -99.00 |
| 2 | Logistic Regression | -95.62 |
| 3 | GaussianNB | -92.61 |
| 4 | K-Nearest Neighbor | -92.02 |



**Figure 4.16 Visual Comparison of Errors predictions between**



**Figure 4.17 Comparison of Model Accuracy**

**Table 4.7 Precision Comparison**

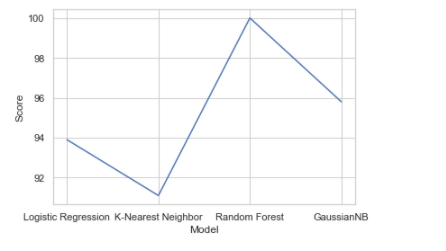
|  |  |  |
| --- | --- | --- |
| **S/n** | **Model Precision** | **Scores** |
| 1 | Random Forest | 100.0 |
| 2 | Logistic Regression | 93 |
| 3 | GaussianNB | 79.0 |
| 4 | K-Nearest Neighbor | 81.0 |



**Figure 4.18 Visual comparison of the model precision**

**Table 4.8** **Comparison of Model Recall / Specificity**

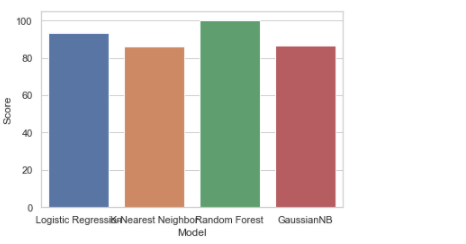
|  |  |  |
| --- | --- | --- |
| **S/n** | **Model Recall** | **Scores** |
| 1 | Random Forest | 100.0 |
| 2 | Logistic Regression | 93.9 |
| 3 | GaussianNB | 95.8 |
| 4 | K-Nearest Neighbor | 91.1 |



**Figure 4.19 Visual comparison of Recall/Specificity for each Model**

**Table 4.9** **Comparison of model F1\_score**

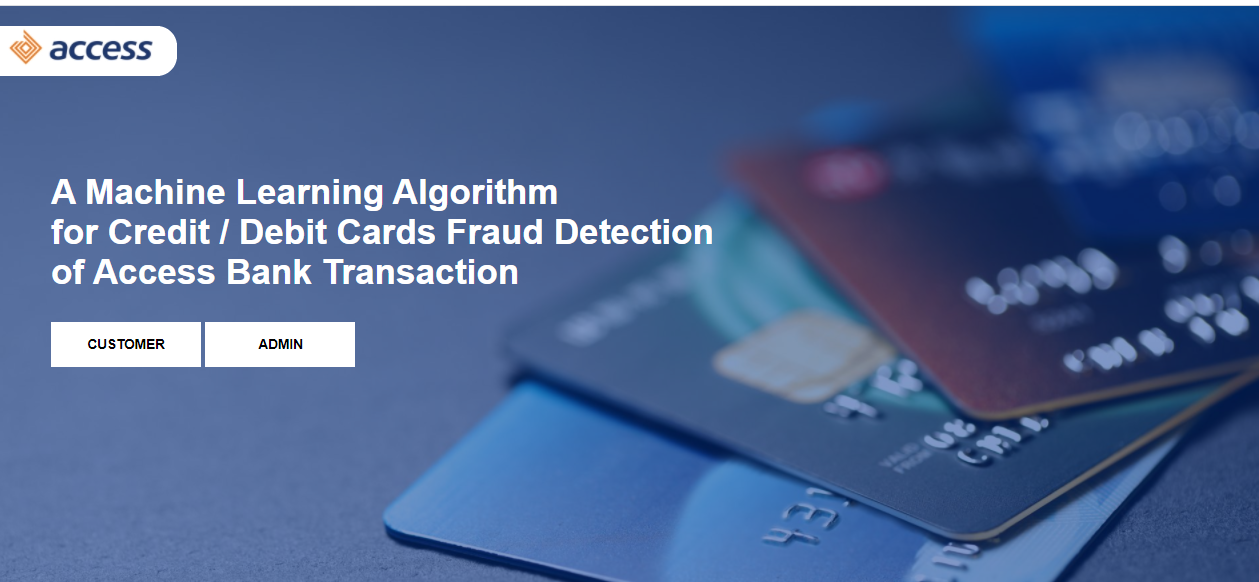
|  |  |  |
| --- | --- | --- |
| **S/n** | **Model Recall** | **Scores** |
| 1 | Random Forest | 100.0 |
| 2 | Logistic Regression | 93.5 |
| 3 | GaussianNB | 86.6 |
| 4 | K-Nearest Neighbor | 85.9 |



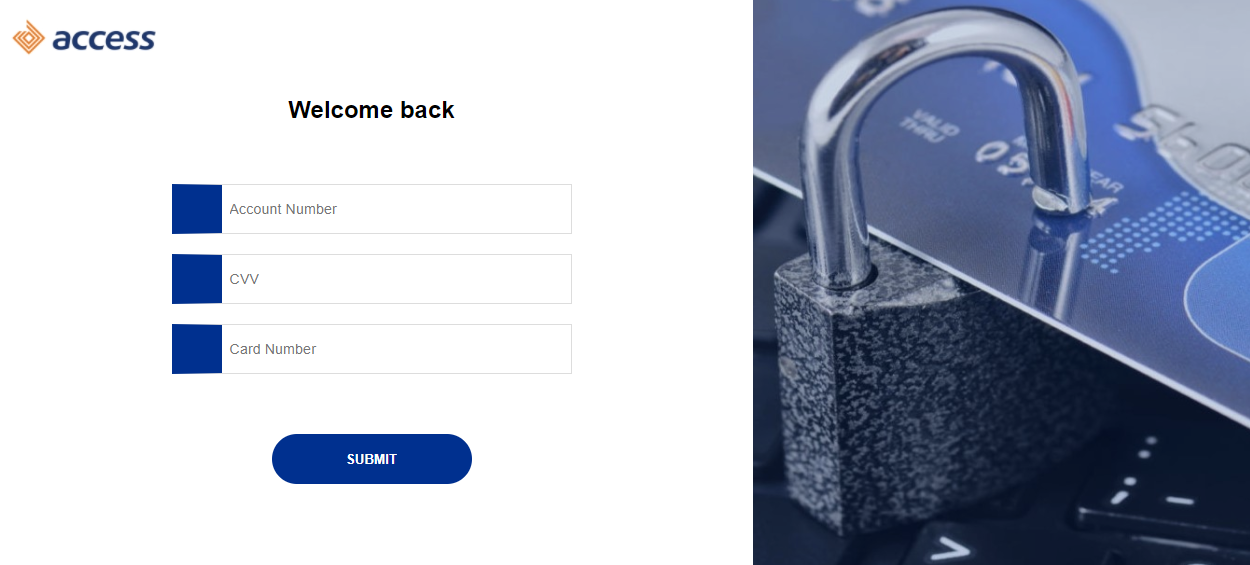
**Figure 4.20 Visual Comparison of F1\_score Model**

**4.8.2 Deployment of Machine Learning Model**

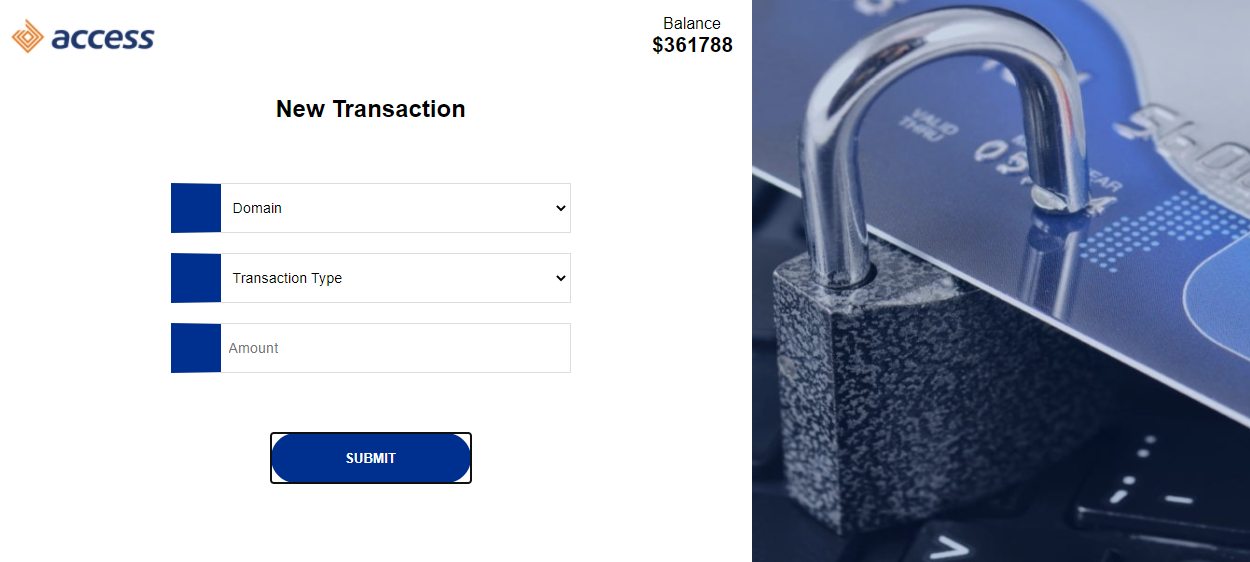
Deployment of machine learning models involves implementing models into production, it also involves making models available to other systems within the organization or the web so that they can receive data as input and return their predictions. The machine learning model was deployed in the web with the aid of the interaction of Python, Cascading Style Sheets and JavaScript. Below is the GUI from the web.



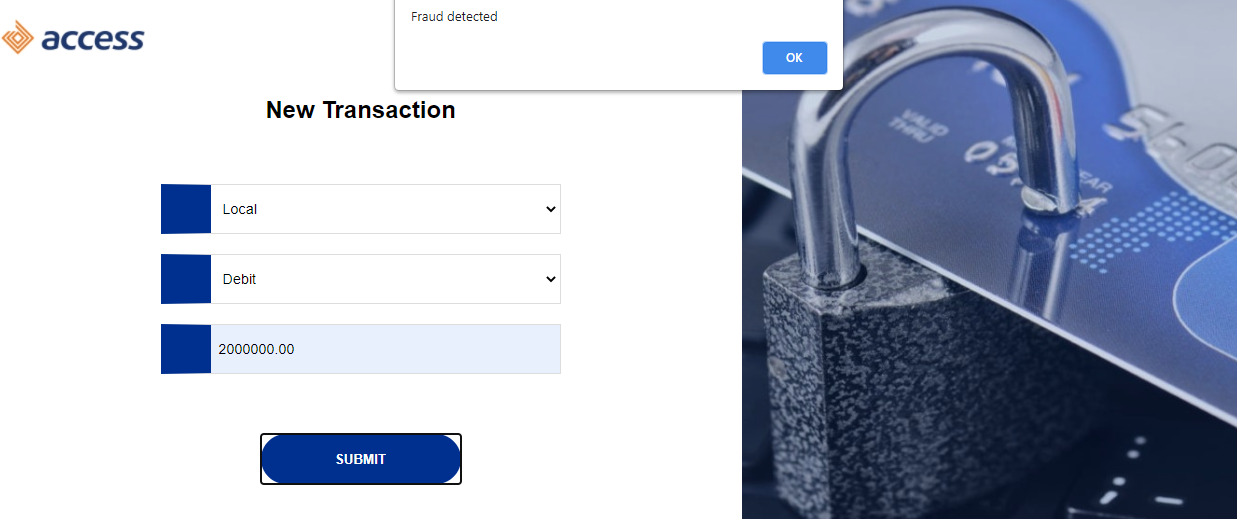
**Figure 4.21 Home page of Graphical User Interface (GUI)**



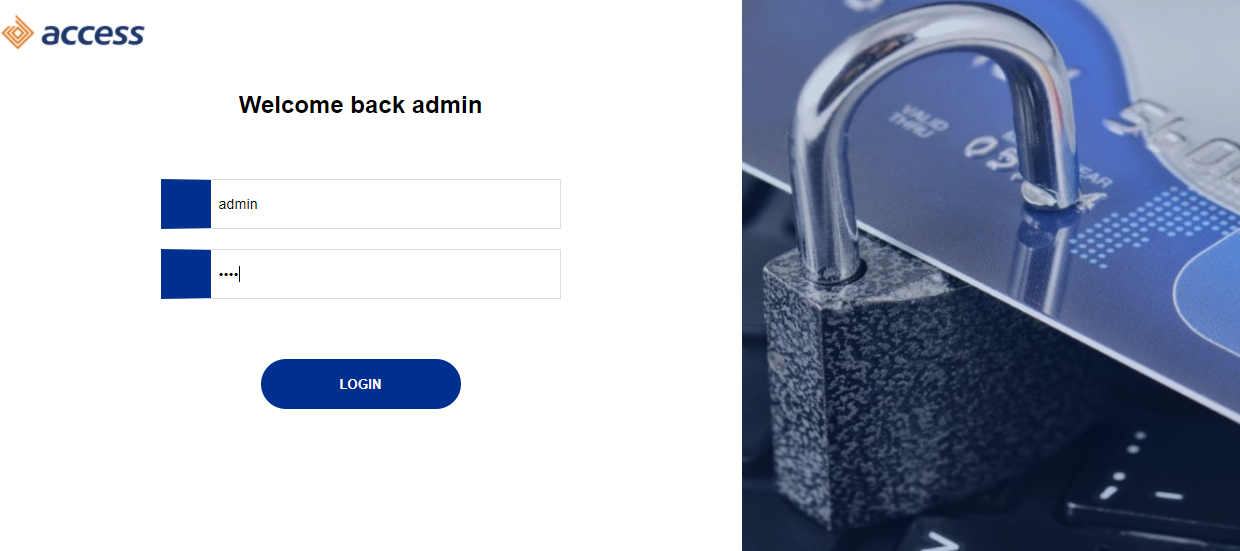
**Figure 4.22 Customer Welcome page**



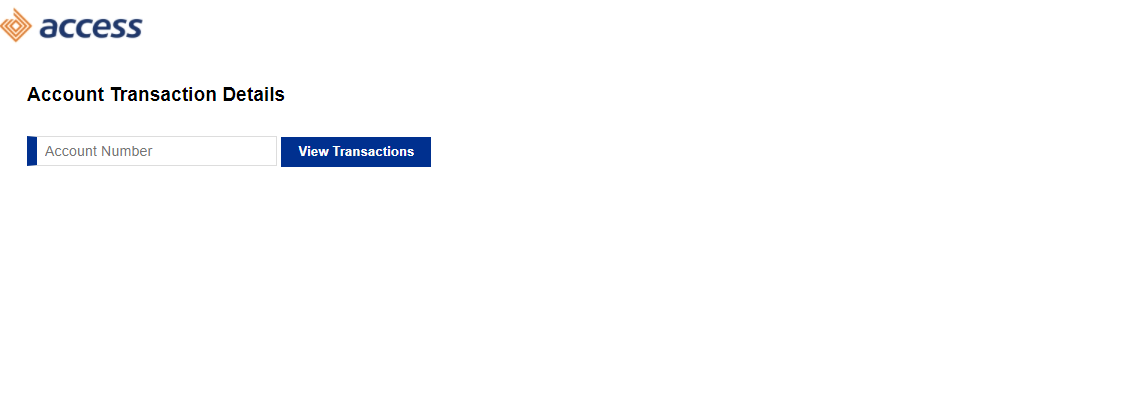
**Figure 4.23 Customers Transaction page**



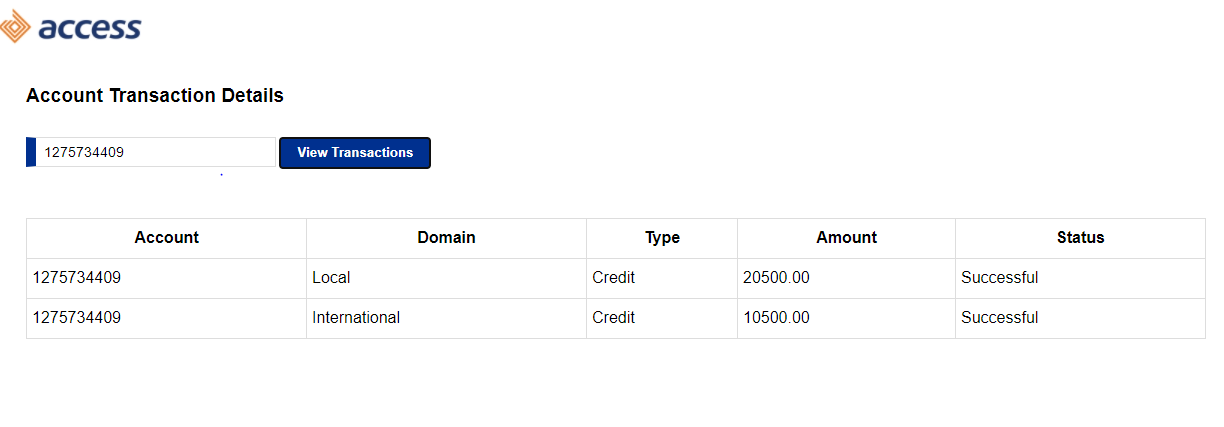
**Figure 4.24 Fraud Transaction detected from a customer transaction**



**Figure 4.25 Administrator Login Page**



**Figure 4.26 Admin Transaction Page**



**Figure 4.27 Customer Transaction from the Admin Page (Tracking transaction at real time)**

**Chapter Five**

**SUMMARY, CONCLUSION AND RECOMMENDATIONS**

* 1. **Summary**

With huge losses in billions of naira and dollars as a result of fraudulent activities and unauthorized access to customer account; there is now more need to secure the system from unauthorized tampering of customers’ accounts. Although technologies and approaches have attempted solving the problem of fraud in credit/debit card transaction systems; serious challenges have plagued the advancement of research in this domain. This includes unavailability of real genuine or fraudulent transaction for dataset for research, the class imbalance problem and problem of concept where a developed model is able to predict new transactions with unfamiliar patterns.

Secondly, the issue of building optimized forests with the Random Forests algorithm is another major problem. Decision trees which form Random Forests must rely on local greedy search to select attributes for splitting. This normally results in local minima and non-optimized trees with the ability to overfit or underfit the learning algorithm resulting in misclassifications. Consequently, this work has suggested a way to solve the problems outlined above by the use of samples of available to real dataset to simulate instances of the real transaction data. The work proposed and implemented an improved algorithm that takes care of correlations and non- correlations in the dataset by the use of Random Forest algorithm for non- correlated data and multivariate data generation.

Similarly, this work implemented an algorithm for genetically evolving and optimizing Random Forests that uses sampling method to build series of random trees, which are randomly used to build forests which generated a higher accuracy than other models.

**5.2 Conclusion**

This work involved the design of a system to detect fraud in credit card transaction with the implementation of machine learning algorithms. This system is capable of providing most of the essential features required to detect fraudulent and legitimate transactions.

Secondly, the dataset was preprocessed and transformed in order to improve the classification accuracy of the various models built by lowering the number of misclassifications and error rate of each prediction. It was also observed that if the dataset is well transformed and well managed from data inconsistency and null values, the dataset can give better optimized model with low overfitting and underfitting rates. This system starts with an initial number of trees which are sampled and grouped into different forests as the starting population. This population of forests is then involved and a better solution capable of lowering training and classification errors chosen as the classification model for the given prediction. This ultimately took care of the class imbalance problem leading to the improved classification accuracy in the developed models.

**5.3** **Recommendations**

From the findings and conclusion stated in this work, the following recommendations are made:

1. Random Forest is recommended for use in all domains of machine learning especially supervised learning where the availability of dataset is a challenge. It is known to handle noisy or missing data as well as categorical or continuous features.

2. Random Forest is also recommended for use in all supervised machine learning problems as it has the potential of giving better classification results against Logistics Regression, K- Nearest Neighbor and Gaussian in terms of accuracy, precision, recall and f1 score.

3. The developed model can be used in a production environment by banks and credit/debit card issuing organizations in order to check incoming transactions and possibly allow or disallow these transactions based on the system decision of the integration and deployment model.

**5.4 Contributions to Knowledge**

This research has contributed the following to knowledge;

1. The design, development and implementation of this work will aid a huge percentage reduction of online fraud transactions.

2. The improvement of classification accuracy and precision of a fraud detection model with the use of Random Forest as against all other forms of classification models.

3. The releasing of synthetically generated fraud datasets to the general public in order to advance research in the field of credit/debit card fraud detection.

**5.5 Suggestion for Further Work**

Although, the algorithm has worked well in our credit/debit card fraud detection system, it is unproven how well this algorithm would perform in order problem domains, especially in domains of large number of features. Getting the correct correlation coefficient between large number of features may as well pose a heavy computation challenge especially with high interdependency between features. Research is need in the application of this model in such research domains.

**References**

[1] E. Nwogu and E. Nwachukwu, “An Improved Hybrid System for the Prediction of Debit and Credit Card Fraud An Improved Hybrid System for The Prediction of Debit and Credit Card Fraud,” no. September, 2019, doi: 10.22624/AIMS/CISDI/V10N3P8.

[2] I. Trivedi and M. Mridushi, “Credit Card Fraud Detection,” vol. 5, no. 1, pp. 39–42, 2016, doi: 10.17148/IJARCCE.2016.5109.

[3] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, “Data mining for credit card fraud: A comparative study,” *Decis. Support Syst.*, vol. 50, no. 3, 2011, doi: 10.1016/j.dss.2010.08.008.

[4] N. Khare and S. Yunus Sait, “Credit Card Fraud Detection Using Machine Learning Models and Collating Machine Learning Models,” *Int. J. Pure Appl. Math.*, vol. 118, no. 20, 2018.

[5] W. W. Soh and R. M. Yusuf, “Predicting Credit Card Fraud on a Imbalanced Data,” *Int. J. Data Sci. Adv. Anal. (ISSN 2563-4429)*, vol. 1, no. 1, 2019.

[6] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, “Credit card fraud detection using machine learning techniques: A comparative analysis,” in *Proceedings of the IEEE International Conference on Computing, Networking and Informatics, ICCNI 2017*, 2017, vol. 2017-January, doi: 10.1109/ICCNI.2017.8123782.

[7] A. Muaz, M. Jayabalan, and V. Thiruchelvam, “A comparison of data sampling techniques for credit card fraud detection,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 6, 2020, doi: 10.14569/IJACSA.2020.0110660.

[8] S. Patil, V. Nemade, and P. K. Soni, “Predictive Modelling for Credit Card Fraud Detection Using Data Analytics,” in *Procedia Computer Science*, 2018, vol. 132, doi: 10.1016/j.procs.2018.05.199.

[9] Z. Zojaji, R. E. Atani, and A. H. Monadjemi, “A Survey of Credit Card Fraud Detection Techniques : Data and Technique Oriented Perspective,” pp. 1–26;2012.

[10] J. R. Prasad, S. Saikumar, and B. V. Subbarao, “Design and development of financial fraud detection using machine learning,” *Int. J. Emerg. Trends Eng. Res.*, vol. 8, no. 9, 2020, doi: 10.30534/ijeter/2020/152892020.

[11] G. Vaughan, “Efficient big data model selection with applications to fraud detection,” *Int. J. Forecast.*, vol. 36, no. 3, 2020, doi: 10.1016/j.ijforecast.2018.03.002.

[12] S. Bagga, A. Goyal, N. Gupta, and A. Goyal, “Credit Card Fraud Detection using Pipeling and Ensemble Learning,” in *Procedia Computer Science*, 2020, vol. 173, doi: 10.1016/j.procs.2020.06.014.

[13] S. K. Shirgave, C. J. Awati, R. More, and S. S. Patil, “A Review On Credit Card Fraud Detection Using Machine Learning,” vol. 8, no. 10, pp. 1217–1220, 2019.

[14] J. Gao, Z. Zhou, J. Ai, B. Xia, and S. Coggeshall, "*Predicting Credit Card Transaction Fraud Using Machine Learning Algorithm"s*, vol. 11, no. 03. 2019.

[15] A. D. Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, “Credit Card Fraud Detection : a Realistic Modeling and a Novel Learning Strategy,” no. October, 2017, doi: 10.1109/TNNLS.2017.2736643.

[16] E. L. Barse, H. Kvarnström, and E. Jonsson, “Synthesizing test data for fraud detection systems,” *Proc. - Annu. Comput. Secur. Appl. Conf. ACSAC*, vol. 2003-Janua, pp. 384–394, 2003, doi: 10.1109/CSAC.2003.1254343.

[17] H. A. Abdou and J. Pointon, “Credit card fraud and detection techniques : A review Credit card fraud and detection techniques : a review,” no. January, 2009.

[18] K. Golmohammadi and O. R. Zaiane, “Data mining applications for fraud detection in securities market,” *Proc. - 2012 Eur. Intell. Secur. Informatics Conf. EISIC 2012*, pp. 107–114, 2012, doi: 10.1109/EISIC.2012.51.

[19] L. Delamaire, H. Abdou, and J. Pointon, “Credit card fraud and detection techniques: A review,” *Banks Bank Syst.*, vol. 4, no. 2, pp. 57–68, 2009.

[20] T. Razooqi, K. Raahemifar, P. Khurana, and A. Abhari, “Credit card fraud detection using fuzzy logic and neural network,” *Simul. Ser.*, vol. 48, no. 3, pp. 47–51, 2016, doi: 10.22360/springsim.2016.cns.009.

[21] S. Ahmed, “Credit Card Fraud Detection using Machine Learning and Data Science,” vol. 8, no. 09, pp. 110–115, 2019.

[22] J. Jurgovsky *et al.*, “Sequence classification for credit-card fraud detection,” *Expert Syst. Appl.*, vol. 100, 2018, doi: 10.1016/j.eswa.2018.01.037.

[23] N. Rtayli and N. Enneya, “Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization,” *J. Inf. Secur. Appl.*, vol. 55, 2020, doi: 10.1016/j.jisa.2020.102596.

[24] F. Carcillo, Y. A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, “Combining unsupervised and supervised learning in credit card fraud detection,” *Inf. Sci. (Ny).*, 2019, doi: 10.1016/j.ins.2019.05.042.

[25] Z. Li, G. Liu, and C. Jiang, “Deep Representation Learning with Full Center Loss for Credit Card Fraud Detection,” *IEEE Trans. Comput. Soc. Syst.*, vol. 7, no. 2, 2020, doi: 10.1109/TCSS.2020.2970805.

[26] E. Kim *et al.*, “Champion-challenger analysis for credit card fraud detection: Hybrid ensemble and deep learning,” *Expert Syst. Appl.*, vol. 128, 2019, doi: 10.1016/j.eswa.2019.03.042.

[27] N. Yousefi, M. Alaghband, and I. Garibay, “A comprehensive survey on machine learning techniques and user authentication approaches for credit card fraud detection,” *arXiv*. 2019.

[28] X. Niu, L. Wang, and X. Yang, “A comparison study of credit card fraud detection: Supervised versus unsupervised,” *arXiv*. 2019.

[29] K. Chandra Kala, N. Ashok Kumar, V. D. Majety, G. L. Sravanthi, and E. Laxmi Lydia, “Constructing a model to predict fraudulent credit card transactions using machine learning techniques,” *Int. J. Adv. Sci. Technol.*, vol. 29, no. 9 Special Issue, 2020.

[30] M. Mohammed, M. B. Khan, and E. B. M. Bashie, *Machine learning: Algorithms and applications*, no. July. 2016.

[31] A. Thennakoon, C. Bhagyani, S. Premadasa, S. Mihiranga, and N. Kuruwitaarac “Real-time credit card fraud detection using machine learning,” 2019, doi: 10.1109/CONFLUENCE.2019.8776942.

[32] R. Saia, L. Boratto, and S. Carta, “Multiple behavioral models: A divide and conquer strategy to fraud detection in financial data streams,” in *IC3K 2015 - Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 2015, vol. 1, doi: 10.5220/0005637104960503.

[33] M. Jeragh and M. Alsulaimi, “Combining Auto Encoders and One Class Support Vectors Machine for Fraudulant Credit Card Transactions Detection,” 2019, doi: 10.1109/WorldS4.2018.8611624.

[34] А. О. Фесенко, Г. К. Папірна, and М. Б. Бауиржан, “The financial fraud detection using machine learning,” *Ukr. Inf. Secur. Res. J.*, vol. 21, no. 2, 2019, doi: 10.18372/2410-7840.21.13769.

[35] N. K. Ahmed, A. F. Atiya, N. El Gayar, and H. El-Shishiny, “An empirical comparison of machine learning models for time series forecasting,” *Econom. Rev.*, vol. 29, no. 5, 2010, doi: 10.1080/07474938.2010.481556.