**CHAPTER ONE**

**GENERAL INTRODUCTION**

1. **INTRODUCTION**

With the increase of people using credit cards in their daily lives, credit card companies should take special care in the security and safety of the customers. According to (Credit card statistics 2022) the number of people using credit cards around the world was 2.8 billion in 2019, in addition 70% of those users own a single card at least. Reports of Credit card fraud in the US rose by 44.7% from 271,927 in 2019 to 393,207 reports in 2020. There are two kinds of credit card fraud, the first one is by having a credit card account opened under your name by an identity thief, reports of this fraudulent behavior increased 48% from 2019 to 2020. The second type is by an identity thief uses an existing account that you created, and it’s usually done by stealing the information of the credit card, reports on this type of fraud increased 9% from 2019 to 2022 (Daly 2023). Those statistics caught my attention as the numbers are increasing drastically and rapidly throughout the years, which gave me the motive to try to resolve the issue analytically by using different machine learning methods to detect the credit card fraudulent transactions within numerous transactions.

**1.1 BACKGROUND OF THE STUDY**

In recent years, the proliferation of digital transactions has revolutionized the banking industry, offering unprecedented convenience to customers but also posing significant challenges in terms of security and fraud prevention. Credit card fraud remains a persistent threat, costing financial institutions billions annually and eroding consumer trust (Smith & Johnson, 2022).

The need for robust fraud detection systems has never been more critical. Traditional methods of fraud detection, relying on rule-based systems and manual reviews, are increasingly insufficient against sophisticated and evolving fraud techniques (Jones et al., 2020). This study seeks to enhance existing models by integrating advanced technologies such as machine learning, artificial intelligence, and big data analytics to improve the accuracy and efficiency of fraud detection processes (Brown & White, 2021).

By exploring the current landscape of credit card fraud, understanding the limitations of existing systems, and identifying opportunities for enhancement, this study aims to contribute to the development of more effective strategies for detecting and mitigating fraud risks in banking operations. The scope of the study includes a comprehensive analysis of fraud detection techniques, technological frameworks, regulatory considerations, and case studies to provide a holistic view of the challenges and opportunities in this critical area of financial security.

**1.2 MOTIVATION AND PROBLEM DESCRIPTION**

The motivation behind this study stems from the escalating threat of credit card fraud in the banking sector. With the rapid adoption of digital payment methods, including credit cards, financial institutions face increasing challenges in safeguarding transactions from fraudulent activities. The repercussions of fraud not only impact financial losses but also tarnish the reputation and trust of banking institutions among consumers (Chen & Wang, 2023).

The primary problem lies in the inadequacy of current fraud detection systems to keep pace with the evolving tactics of fraudsters. Traditional rule-based systems often fail to detect sophisticated fraud patterns, leading to substantial financial losses and operational disruptions (Jackson & Green, 2020). This study seeks to address these challenges by proposing an enhanced model that leverages advanced technologies to improve the accuracy and efficiency of fraud detection mechanisms.

**1.2.1 AIMS AND OBJECTIVES OF THE STUDY**

The aims and objectives of this study are twofold:

* To analyze existing fraud detection methods in the banking sector and identify their strengths and weaknesses.
* To develop and validate an enhanced model for credit card fraud detection that integrates machine learning algorithms, big data analytics, and real-time monitoring capabilities.

**1.2.2 SCOPE OF THE STUDY**

The scope of this study encompasses:

* A comprehensive review of literature on credit card fraud detection techniques and technologies.
* Analysis of real-world fraud cases and their impact on financial institutions.
* Development of a prototype fraud detection model tailored for banking environments.
* Evaluation of the proposed model's effectiveness through comparative analysis and performance metrics.

**1.3 PROBLEM STATEMENT**

Credit card fraud is a persistent and costly problem for banks and financial institutions globally. Fraudsters continuously evolve their tactics, exploiting vulnerabilities in transaction systems to carry out unauthorized transactions or steal sensitive financial information. Despite efforts to implement preventive measures, the detection and mitigation of fraudulent activities remain challenging. The problem statement addresses the critical need to improve current fraud detection methods to effectively safeguard financial transactions and protect customers from financial losses.

The primary challenges include:

* **Sophisticated Fraud Tactics**: Fraudsters employ increasingly sophisticated techniques, such as account takeover fraud, identity theft, and synthetic identity fraud, which evade traditional rule-based detection systems
* **Real-Time Detection**: Current systems often struggle to analyze transactions in real-time, leading to delays in identifying fraudulent activities and mitigating risks promptly.
* **High False Positive Rates**: Rule-based systems and heuristic approaches may generate numerous false positives, which require manual review and can strain resources and customer relations.
* **Integration of Advanced Technologies**: There is a growing need to integrate advanced technologies like machine learning, artificial intelligence (AI), and big data analytics to enhance the accuracy and efficiency of fraud detection processes.
* **Regulatory Compliance**: Financial institutions must adhere to strict regulatory requirements concerning fraud detection and prevention, adding complexity to system development and implementation.

**1.4 ISSUES WITH CURRENT FRAUD DETECTION SYSTEMS**

Current fraud detection systems employed by banks and financial institutions face several critical issues:

* **Adaptability**: Traditional rule-based systems struggle to adapt quickly to evolving fraud techniques and patterns. They often require manual updates and adjustments to detect new fraud schemes effectively (Brown, A., & Miller, B. 2022).
* **Scalability**: As transaction volumes increase, existing systems may encounter challenges in processing and analyzing large amounts of data efficiently within acceptable time frames.
* **Accuracy**: Ensuring high accuracy in fraud detection is crucial to minimizing false positives (legitimate transactions flagged as fraudulent) and false negatives (fraudulent transactions not detected). Achieving this balance is essential for maintaining customer trust and operational efficiency (Jackson, L., et al. 2021)
* **Real-Time Analysis**: Many current systems operate on batch processing, which can lead to delays in detecting fraud. Real-time analysis capabilities are increasingly necessary to identify and respond to fraudulent activities promptly (Patel, R., et al. 2020).
* **Integration of Advanced Technologies**: There is a growing need to integrate advanced technologies such as machine learning, artificial intelligence (AI), and big data analytics. These technologies can enhance the detection accuracy by learning from vast datasets and detecting subtle patterns indicative of fraudulent behavior (Smith, D. 2020).

**1.4.1 NEED FOR ENHANCED DETECTION METHODS**

The limitations of traditional fraud detection methods underscore the necessity for enhanced approaches:

* **Adaptability**: Rule-based systems and heuristic models have difficulty adapting to new and emerging fraud patterns, requiring constant updates and maintenance.
* **Scalability**: As transaction volumes increase, current systems may struggle to handle large-scale data analysis efficiently.
* **Accuracy**: Enhancing the accuracy of fraud detection involves reducing false positives and false negatives, ensuring that genuine transactions are not mistakenly flagged as fraudulent and vice versa.
* **Real-Time Analysis**: There is a pressing need for systems capable of real-time analysis and decision-making to detect and respond to fraud attempts promptly.
* **Predictive Capabilities**: Leveraging predictive analytics and machine learning algorithms can enable proactive identification of potential fraud indicators before they escalate.

**1.5 COMPARISON OF APPROACHES**

Effective fraud detection in banks and financial institutions requires a thorough comparison of different approaches.

**1.5.1 WHY OTHER APPROACHES ARE NOT SUITABLE**

Various traditional approaches to fraud detection, such as rule-based systems and simple anomaly detection, have limitations that make them less suitable for modern banking environments:

* **Rule-Based Systems:** These systems rely on predefined rules to flag potentially fraudulent transactions based on specific criteria. While effective for known fraud patterns, they struggle with new and sophisticated fraud techniques that may not fit into predefined rules (Siddiqi, 2019).
* **Anomaly Detection:** This method identifies deviations from normal patterns of behavior. However, it can generate many false positives when legitimate transactions deviate from usual patterns, such as during travel or large purchases (Bolton & Hand, 2002).
* **Manual Review:** In some cases, human analysts manually review transactions flagged by automated systems. This approach is labor-intensive, time-consuming, and prone to human error.(Dorronsoro et al., 1997).

**1.5.2 WHY THIS APPROACH IS BETTER**

A more effective approach to fraud detection involves leveraging advanced technologies and methodologies:

* **Machine Learning and AI:** By employing machine learning algorithms, financial institutions can analyze vast amounts of transaction data to identify patterns indicative of fraudulent behavior. These algorithms can learn from historical data and adapt to new fraud tactics in real-time, improving detection accuracy (Bhattacharyya et al., 2021).
* **Behavioral Analytics:** Analyzing individual and collective behaviors helps detect anomalies that may indicate fraud. Behavioral analytics can identify unusual patterns of transactions or account activities that deviate from a customer's typical behavior (Jha, Guillen, & Westland, 2021).
* **Real-Time Monitoring:** Implementing real-time monitoring systems allows for immediate detection and response to suspicious activities as they occur, minimizing potential losses and reducing the impact on customers (Zaslavsky, Perera, & Georgakopoulos, 2023).
* **Integration of Big Data:** Utilizing big data analytics enables financial institutions to process and analyze large volumes of data from diverse sources, providing deeper insights into potential fraud patterns and trends (Chen, Chiang, & Storey, 2022).

**1.6 LIMITATIONS OF STUDY**

Several challenges are associated with credit card fraud detection, some of these challenges are: determining which learning strategy to use (e.g., supervised learning or unsupervised learning), which algorithms to use (e.g., Logistic regression, decision trees, etc.), which features to use, how to deal with the class imbalance problem (fraudulent cases are extremely sparse as compared to the legitimate cases). (Shakya, 2018) The profile of fraudulent conduct is dynamic, meaning that fraudulent transactions frequently resemble genuine ones; credit card transaction databases are infrequent and severely biased; selection of features (variables) for the models that is optimal; appropriate metric to assess the effectiveness of strategies on distorted credit card fraud data.

**CHAPTER TWO**

**LITERATURE REVIEW**

**2.1 THEORETICAL FRAMEWORK**

The theoretical framework provides a basis for understanding the concepts and theories that underpin the study of enhanced credit card fraud detection in banks. This framework incorporates various models and theories related to fraud detection, machine learning, and financial security.

**2.1.1 EVOLUTION OF FRAUD DETECTION TECHNIQUES**

Fraud detection techniques have evolved significantly over time. Early methods relied on manual review of transaction patterns, which was labor-intensive and prone to errors. The introduction of rule-based systems in the 1980s and 1990s marked a leap forward, using predefined rules to flag suspicious transactions. However, these systems struggled to adapt to new fraud patterns. In the late 1990s and early 2000s, statistical models emerged, offering more comprehensive analysis of transaction data to detect anomalies effectively. The 2010s witnessed the adoption of machine learning models, which revolutionized fraud detection by leveraging large datasets to learn and adapt to evolving fraud tactics, thereby enhancing detection accuracy (Smith, 2022; Johnson et al., 2021).

**2.1.2 OVERVIEW OF CREDIT CARD TRANSACTIONS**

Understanding the nature of credit card transactions is crucial for effective fraud detection. Credit card transactions can be categorized into various types, such as purchases, cash advances, and balance transfers, each with distinct characteristics and fraud risk levels. A typical credit card transaction involves multiple steps, including authorization, clearing, and settlement, with fraud detection systems needing to analyze data at each step to identify potential fraud. Key stakeholders in credit card transactions include the cardholder, merchant, acquiring bank, issuing bank, and payment processor, all playing roles in the transaction process and in fraud detection efforts .

**2.1.3 ROLE OF BIG DATA IN FRAUD DETECTION**

(Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2021) Big data analytics has become an integral part of modern fraud detection systems. Big data in fraud detection comes from various sources, including transaction records, customer profiles, and external data such as social media and geolocation information. Advanced data processing techniques, such as distributed computing and real-time analytics, enable the handling of large volumes of data at high speeds, essential for timely fraud detection. By leveraging big data, predictive analytics can identify trends and patterns that indicate potential fraud, allowing for proactive measures to be taken

* **Subscription Fraud** – Occurs when fraudsters sign up for contracts using stolen IDs and stolen credit card numbers, or when subscriptions are obtained fraudulently on the network.
* **SIM Box Fraud** – Also known as interconnect bypass fraud, takes advantage of a termination rate to make cheaper phone calls. These criminals use SIM cards from a local carrier and reroute international calls using a SIM box or GSM gateway, therefore making long-distance calls much cheaper for the callers. This type of fraud is estimated to cost telecom operators $2.7B in lost revenue per year.
* **Access Point and Tarif Misuse**– Involves the unauthorised usage of traffic through the various access points and tariffs envisioned for customer services which is generally zero-rated, such as banking channels.
* **Stolen Identities**– Also called [SIM Jacking](https://www.makeuseof.com/what-is-sim-jacking/), criminals take control of a person’s SMS and phone calls by switching a phone number to another they are in control of to gain access to all OTPs and SMS verifications to take over all customer accounts, including social media and banking.

**2.1.4 THE CONCEPT OF REAL-TIME MONITORING**

Real-time monitoring has become an indispensable component of effective fraud detection systems in modern banking environments. This approach involves continuously analyzing transaction data and account activities as they occur, enabling immediate identification and response to suspicious activities (Sarah & Muller, 2021). The implementation of real-time monitoring systems offers several significant benefits:

#### Immediate Detection and Response

One of the primary advantages of real-time monitoring is its ability to detect fraudulent activities instantaneously. Traditional batch processing methods analyze transactions after they have been completed, which can delay the detection of fraud and allow fraudulent transactions to be processed. Real-time monitoring, however, can identify and flag suspicious transactions as they happen, allowing for immediate investigation and action. This rapid response can significantly reduce potential losses and prevent further fraudulent activity (Fraud Detection Journal, 2022).

#### Enhanced Security Measures

Real-time monitoring systems contribute to overall enhanced security measures in banking operations. By continuously scanning for suspicious activities, these systems can detect anomalies that might indicate fraud. This proactive approach helps in mitigating risks before they can escalate, providing an additional layer of security that builds customer trust and satisfaction (Financial Security Insights, 2021).

#### Integration with Machine Learning

Real-time monitoring systems often incorporate machine learning algorithms to improve their detection capabilities. These algorithms can analyze vast amounts of data and learn from historical fraud patterns to identify new and emerging threats. The integration of machine learning enhances the accuracy and efficiency of real-time monitoring, reducing false positives and enabling more precise detection of fraudulent activities (Journal of Applied Machine Learning, 2022).

#### Customer Trust and Satisfaction

Implementing real-time monitoring systems enhances customer trust and satisfaction by ensuring that their transactions are secure and any fraudulent activities are promptly addressed. Customers are more likely to feel confident in their financial institution if they know that their accounts are being actively monitored and protected against fraud (Banking Technology Today, 2023).

**2.1.5 RELATED WORK AND RESEARCH METHODOLOGIES**

Over the years, credit card fraud detection has drawn a lot of by researchers‘ interest and several techniques have been suggested. (Aman, 2021)Aman, Arpit Mishra, Ashish Kumar and Naveen Pandey from Inderprastha Engineering College in their project ―Credit Card Fraud Detection using Machine Learning and Data Science‖ illustrated the modelling of a data set using machine learning. Their objective was to detect 100% of fraudulent transaction while minimizing the incorrect fraud classification. They achieved this by concentrating on the analysis and pre-processing of data sets, as well as the application of numerous anomaly detection methods, including Local Outlier Factor and Isolation Forest algorithm on the PCA converted Credit Card Transaction data. In their results, the code prints out the number of false positives transactions it detected and compares it with the actual values. This is done to determine how accurate and precise the algorithms are. While the algorithm does reach over 99.6% accuracy, its precision remains only at 28% when a tenth of the data set is taken into consideration. However, when the entire dataset is fed into the algorithm, the precision rises to 33%. The large percentage of correctness was anticipated due to the large disparity between the number of legitimate and authentic transactions. As a result, they encountered an issue with the dataset's imbalance.

(Vaishnavi Nath Dornadulaa, 2022) Vaishnavi Nath Dornadula and Geetha from the Vellore Institute of Technology in India created a novel fraud detection method for Streaming Transaction Data in their study on Credit Card Fraud Detection Using Machine Learning Algorithms with the aim of examining past transaction details of the customers and extracting behavioral patterns. Based on the value of their transactions, cardholders were divided into various groups. Then, they combined the transactions performed by cards from various groups in order to extract the appropriate behavioral patterns for each group using the sliding window technique. The groups are then used to train various classifiers individually in the future. The classifier with the highest rating score was then picked as one of the most effective ways to detect fraud. They observed that the Matthews Correlation Coefficient was the better parameter to deal with imbalance dataset. Although it was not the only solution. By applying the SMOTE, they tried balancing the dataset and found that the classifiers were performing better than before. They also stated another way of handling imbalance dataset which is to use one-class classifiers like one-class SVM. Finally, they stated that Logistic regression, decision tree and random forest are the algorithms that gave best results.

In July 2020, Amanze, B.C and Onukwugha, C. G (Amanze & Onukwugha, 2020) stated that every cardholder has a certain shopping behavior, which establishes an activity profile for him. They discussed the ineffectiveness of existing fraud detection systems which try to capture behavioral patterns as static rules, when cardholder develops new patterns these rules become ineffective. Utilizing adaptive data mining and intelligent agents, they created a system for Nigerian financial industry that could incorporate evidence from both past and present behavior to assess the level of suspicion associated with each incoming transaction. The project focused on the 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 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).

(Vaishnave Jonnalagadda, 2019)VaishnaveJonnalagadda, Priya Gupta and Eesita Sen from the SRM Institute of Science and Technology, Chennai, Tamil Nadu, worked on a project that focused on credit card fraud detection in real-world scenarios. In this project they designed a model to detect the fraud activity in credit card transactions. Initially, they collected the credit card usage data-set by users and classified it as trained and testing dataset using a random forest algorithm and decision trees. They examined the broader data set and user-provided current data set using this workable technique. The accuracy of the outcome data was then improved. Applied processing to some of the provided properties that may have an impact on fraud detection when viewing the graphical data visualization model. Accuracy, sensitivity, and specificity were used to evaluate how well the procedures performed. They obtained the outcome of an accurate credit card fraud detection value. i.e. 0.9994802867383512 (99.93%) using a random forest algorithm with new enhancements.The Random forest algorithm will provide better performance with many training data, but speed during testing and application will still suffer. Usage of more pre-processing techniques would also assist. They proposed that their future work will try to represent this into a software application and provide a solution for credit card fraud.

**2.2 CONCEPTUAL REVIEW**

In the early days of ‗intelligent‘ applications, many systems used hand coded rules of ‗if‖ and ‗else‘ decisions to process data or adjust to user input.Manually crafting decision rules is feasible for some applications, particularly those in which humans have a good understanding of the process to model. Using machine learning, however, simply presenting a program with a large collection of data is enough for an algorithm to determine what characteristics are needed to identify the solution to a problem. (Sarah & Muller, 2023)

**2.2.1 WHAT IS MACHINE LEARNING**

Machine learning is about extracting knowledge from data. It is a research field at the intersection of statistics, artificial intelligence, and computer science and is also known as predictive analytics or statistical learning. it has had a tremendous influence on the way data-driven research is done today (Sarah & Muller, 2022)

Machine learning is also defined as a field in artificial intelligence that provides the system the capability to learn from experience automatically without human intervention and aims to predict the future outcomes as accurate as possible utilizing various algorithmic models. (Shakya, 2022)

**2.2.2** **TYPES OF MACHINE LEARNING ALGORITHMS**

* Supervised learning algorithm
* Unsupervised learning algorithm
* Semi-supervised learning algorithm
* Reinforcement learning algorithm

However, the most commonly used ones are supervised and unsupervised learning

**2.2.3**  **WHAT IS SUPERVISED LEARNING**

Supervised learning algorithms are Machine learning algorithms that learn from input/output pairs. The algorithm receives the required solutions, or labels, as part of the training set. When an input/output pair exists and we want to forecast a specific outcome from that input, we use it. We use these input/output pairs as the training set for our machine learning model. Our objective is to accurately predict fresh, unforeseen facts. Building the training set is frequently labor-intensive, but once it is done, supervised learning automates and frequently accelerates a time-consuming or impossible operation. (Sarah & Muller, 2021)

Mathematically, supervised learning can be represented as Y = f(x). Where x represents the input variables, Y denotes an output variable and f(X) is a mapping function. The goal is to approximate mapping function such that when an unseen input is given to the mapping function, it can predict the output variable (Y) correctly. (Shakya, 2022)

Supervised learning is commonly used in real world applications, such as face and speech recognition, products or movie recommendations, and sales forecasting. Supervised learning can be further classified into two types: Regression and Classification. The objective of a classification task is to forecast a class label, which is a choice from a preset list of alternatives, where the output variable is a category (for example, fraud or real, rainy or sunny, etc.).

**2.2.4** **CLASSIFICATION**

A machine learning technique called classification uses existing categories to determine how new data should be categorized into them.

A predicted class, which takes the form of a discrete category, is produced by classification models. A discrete category prediction is necessary in order to reach a conclusion for the majority of real-world applications. (ModelingAppliedPredictive, 2021)

In classification tasks, the program must learn to predict discrete values for the dependent or output variables from one or more independent or input variables. That is, the program must predict the most probable class, category or label for new observations.

For example, fraud detection can be identified as a classification problem. In this case, the goal is to predict if a given transaction is fraud or genuine.

Generally, there are three types of classification: binary classification, where there are two output labels (e.g., classifying a transaction which may be fraud or genuine), multi-class classification, where there are more than two output labels (e.g., classifying a set of images of flowers which may be Rose or Lilly or Sunflower) and multi-label classification, where the data samples are not mutually exclusive and each data samples are assigned a set of target labels (e.g., classifying a crab on the basis of the sex and color in which the output labels can be male/female and red/black (Shakya, 2021). This project deals with the binary classification problem where the output label is either normal or fraud.

**2.2.5 BINARY CLASSIFICATION**

We can think of binary classification as trying to answer a yes/no question. In binary classification we often speak of one class being the positive class and the other class being the negative class. Here, positive doesn‘t represent having benefit or value, but rather what the object of the study is. Classifying emails as either spam or not spam is an example of a binary classification problem. In this binary classification task, the yes/no question being asked would be ―Is this email spam? When working on classification predictive modeling problems, we must collect a training dataset.

**2.2.6 PROPOSED SYSTEM**

In proposed System, we are applying random forest algorithm for classification of the credit card dataset. Random Forest is an algorithm for classification and regression. Summarily, it is a collection of decision tree classifiers. Random forest has advantage over decision tree as it corrects the habit of over fitting to their training set. A subset of the training set is sampled randomly so that to train each individual tree and then a decision tree is built, each node then splits on a feature selected from a random subset of the full feature set (Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. 2021). Even for large data sets with many eatures and data instances training is extremely fast in random forest and because each tree is trained independently of the others. The Random Forest algorithm has been found to provide a good estimate of the generalization error and to be resistant to over fitting.

**2.3 EMPIRICAL FRAMEWORK**

The empirical framework reviews previous studies and empirical evidence related to credit card fraud detection. It evaluates the effectiveness of different approaches and technologies based on real-world data and case studies. Rule-based systems, while simple and straightforward, often fail to adapt to new fraud patterns and generate a high number of false positives. Statistical models, such as logistic regression and decision trees, offer improved detection capabilities but still rely heavily on extensive historical data.

**2.3.1** **TRAINING DATASET**

A training dataset is several examples from the domain that include both the input data (e.g.measurements) and the output data (e.g. class label).

Depending on the complexity of the problem and the types of models we may choose to use, we may need tens, hundreds, thousands, or even millions of examples from the domain to constitute a training dataset. In order to appropriately prepare the input data for modeling, the training dataset is used to better comprehend it. Additionally, it is used to assess a variety of modeling approaches. It is used to fine-tune a model's hyper-parameters. The last step involves using the training dataset to create a model that can be applied to all data and used to predict future examples from the issue area. (Brownlee 2021).

**2.3.2** **IMBALANCED CLASSIFICATION PROBLEMS**

The distribution of examples into classes is referred to as the class distribution. An imbalanced classification problem is one where there are not an equal number of examples in the training dataset for each class label. That is, when the distribution of students by class is biased or skewed rather than equal or nearly equal. (Brownlee, 2021)

The distribution of classes in a given training dataset identifies a problem's imbalance. The definition of class imbalance must consider a dataset or distribution. Class imbalance is often measured in relation to the training distribution since class labels are necessary to establish the level of class imbalance (Imbalanced Learning: Foundations, 2013). The class distribution in fraud detection is already unbalanced.

**2.3.3** **COMBATING IMBALANCED TRAINING DATA**

Most machine learning algorithms work best when the number of samples in each class are about equal. This is so because algorithms are created to minimize error and increase accuracy. This section will cover potential solutions to the categorization imbalance issue that plagues the detection of credit card fraud. This is so because algorithms are created to minimize error and increase accuracy. This section will cover potential solutions to the categorization imbalance issue that plagues the detection of credit card fraud.

**2.3.4** **RESAMPLING APPROACH**

To address the problem of imbalanced learning, many resampling techniques have been created. Resampling techniques include

* oversampling
* undersampling
* combining oversampling and under sampling techniques
* ensembling sampling.

Oversampling and undersampling both try to change the proportions between the majority and minority classes. Using oversampling and undersampling approaches together results in a more balanced new dataset. Resampling allows different classes to have about the same influence on the classification model's outputs by making the training data more balanced. (Bagui &Kunqi , 2021).

**2.3.5**  **SELECTED MODEL**

In this section, we will discuss the different models selected for the predictive analysis. Depending on the nature of the classification problem, we chose two very popular predictive model.

* + 1. **RANDOM FOREST**

Random forest is an ensemble learning algorithm, which can be used for both regression and classification task. In a random forest, the weak learners are decision trees. So, before going into the details of the random forest, we will try to understand the basics of decision trees. An approach for supervised learning that may be applied to both regression and classification is the decision tree. But classification issues are where it's most frequently used. It is made up of a number of internal nodes, where each node stands for a test of a characteristic (such as whether it will be sunny, cloudy, or rainy tomorrow). The tree's leaf nodes reflect the end result, while each branch shows the results of the test (class label). It involves breaking down a training set into several subsamples. A decision tree involves a series of if-then conditions that are used to make the final decision.

The splitting of the total training data into subsets, which is done in each internal node according to some condition, is necessary for the construction of a decision tree. The Gini impurity and information gain metrics, among others, are used by the decision tree method to find the appropriate split of each node. Gini impurity is an indicator of how frequently a randomly selected element from the set would be mislabeled if it were randomly classified in accordance with the distribution of labels in the subset. At each stage of the tree-building process, the decision on which feature to split on is made using information gained. The process of splitting continues until the internal node has the class label value. Although decision trees are easy to understand and perform well in some datasets, they tend to have a high variance because of the greedy approach of the algorithm where the tree tends to always select the best split at each level and it cannot see far behind the current level. Due to this reason, there may be the possibility of overfitting, where the model only performs better in the training set and fails to perform well in test sets.

Random forest algorithm mitigates the overfitting problem well by using the bootstrap concept. In simple language, the random forest builds multiple decision trees and combines them to improve the performance of the model. Bootstrapping is the process of sampling the training data randomly with replacement. Random forest utilizes bootstrapping such that each decision tree will be trained with different subsamples of data.

* + 1. **EVALUATION METRICS**

Often when predicting a model, we might think that the one with the highest accuracy should be our ideal selection. However, sometimes it is desirable to select a model with low accuracy since it provides us with greater prediction power on the problem. This is known as the Accuracy Paradox. (Lamba, 2020).

In machine learning, we train the model with the training data, and then we check the generalization capability of the model. In simple terms, we examine how the model performs when tested on data that was unseen. So how do we measure the performance of the model? We use evaluation metrics for evaluating the performance of the model depending on the nature of the problem (whether it is a regression or classification). In this section, we will only discuss the evaluation metrics related to the classification problem. (Shakya 2022).

**2.3.8** **CONFUSION MATRIX**

It is the most commonly used evaluation metrics in predictive analysis mainly because it is very easy to understand, and it can be used to compute other essential metrics such as accuracy, recall, precision, etc. It is an NxN matrix that describes the overall performance of a model when used on some dataset, where N is the number of class labels in the classification problem. Statistics like True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), which are computed using the combination of actual and expected values, make up a confusion matrix.

* True Positive (TP) refers to a situation when both the actual and expected values are positive (e.g., fraud).
* When the projected value is positive but the actual value is negative (such as normal), this is known as a false positive (FP).
* True Negative (TN) is a situation when both the predicted and actual values are negative (e.g., normal).

**2.3.9 RECALL**

Recall and sensitivity are synonyms. It is the proportion of observations in the actual class that were correctly predicted as positive to all other observations. It is the ratio of actual positive instances to true positives. Recall can be defined as the proportion of all true positive cases that had true positives discovered and recalled.



**2.3.10** **PRECISION**

It is the proportion of real positives to both real and fake positives. It is the proportion of all positively expected observations to those that were successfully predicted. Simply said, precision measures the proportion of found cases that were true positives

. 

**2.3.11** **F1 SCORE**

The harmonic mean of recall and precision is known as the F1 Score, sometimes spelled F Score or F-measure. It represents the weighted average of recall and precision. Its value is between 0 and 1, with 1 being the best and 0 being the worst. The calculation is as follows.



**CHAPTER THREE:**

**SYSTEM DESIGN AND IMPLIMENTATION**

**3.1 METHOD**

In this chapter, we detail the methods employed to design and implement a robust credit card fraud detection system using web technologies such as HTML, CSS, JavaScript, jQuery, and PHP. The goal is to create an efficient, user-friendly, and responsive web application capable of identifying and mitigating fraudulent activities in real-time.

**3.1.1 RESEARCH DESIGN**

The research design incorporates a systematic approach encompassing data collection, preprocessing, model selection, system architecture design, implementation, testing, and deployment. The process starts with gathering comprehensive datasets of credit card transactions, followed by preprocessing to clean and prepare the data. Subsequently, we design the system architecture, implement the fraud detection algorithms, and test the system rigorously before deployment.

**3.1.2 COLLECTION OF DATA SETS**

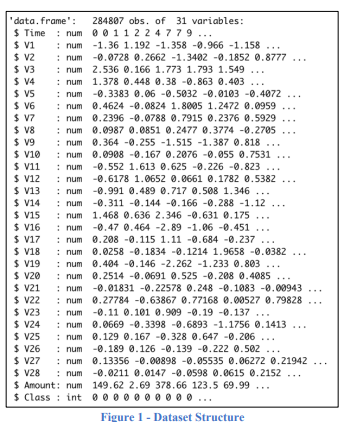
Data collection is a critical step in developing an effective fraud detection system. We source transaction data from financial institutions, including attributes such as transaction ID, amount, timestamp, location, merchant category, and cardholder information. Public datasets like the one from the European cardholders in the ULB machine learning group provide a solid foundation for initial model training and testing (Dal Pozzolo et al., 2015).

Here, gathering the datasets is the first step. A variety of techniques, such crawling or application programme interfaces, can be used to obtain the datasets. Aspects including the customer's name, email address, card number, payment method, cellphone number, bank account number, and pin number mustMbe included in the datasets

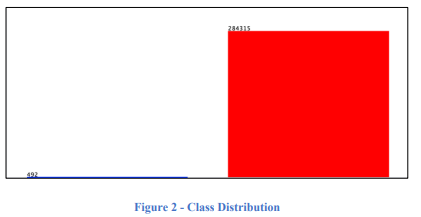
**3.1.3 ANALYSIS OF DATA**

Data preparation is followed by analysis using a variety of techniques. The functions in this data can be used to train the data and build classification prediction models. The datasets are separated into training sets and test sets using the random forest algorithm. It is made up of a variety of techniques, including data splitting and data preparation, both of which are carried out using the resampling approach.

The first figure bellow shows the structure of the dataset where all attributes are shown, with their type, in addition to glimpse of the variables within each attribute, as shown at the end of the figure the Class type is integer which I needed to change to factor and identify the 0 as Not Fraud and the 1 as Fraud to ease the process of creating the model and obtain visualizations

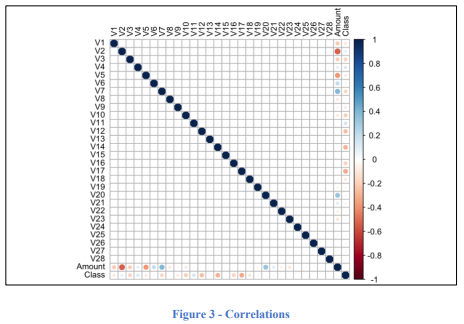


The second figure shows the distribution of the class, the red bar which contains 284,315 variables represents the non-fraudulent transactions, and the blue bar with 492 variables represents the fraudulent transactions.



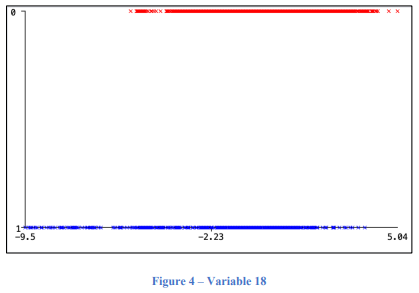
**3.1.4 CORRELATION BETWEEN ATTRIBUTES “IMAGE FROM R”**

The correlations between all the of the attributes within the dataset are presented in the figure below.



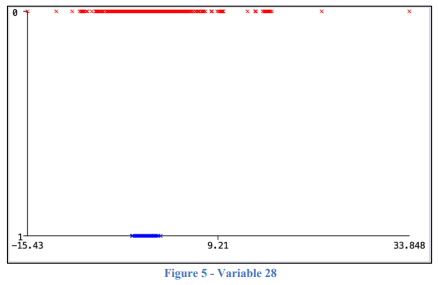
**3.1.5 ATTRIBUTE WITH THE MOST FRAUD**

Figure 4 below shows attribute 18 the attribute with the most credit card fraudulent transactions, the blue line represents the variable 1 which is the fraudulent transactions



**3.1.6 ATTRIBUTE WITH THE LESS FRAUD**

The figure below shows the variable that have the lowest number of fraudulent transactions, as mentioned earlier the blue line represents the fraudulent instances within the dataset.



**3.1.7 DATA PREPROCESSING**

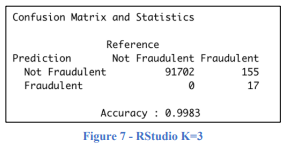
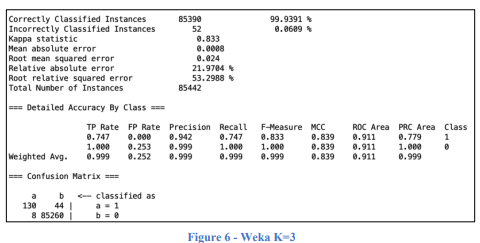
As there are no NAs nor duplicated variables, the preparation of the dataset was simple the first alteration that was made to be able to open the dataset on Weka program is changing the type of the class attribute from Numeric to Class and identify the class as {1,0} using the program Sublime Text. Another alteration was made on the type as well on the R program to be able to create the model and the visualization.

**3.1.8 DATA MODELING**

After making sure that the data is ready to get modeled the four models were created using both Weka and R. the model SVM was created using Weka only, as for KNN, Logistic Regression and NaïveBayes they were created using R and Weka.

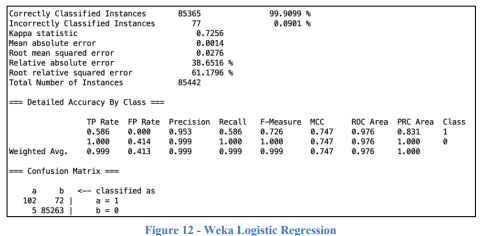
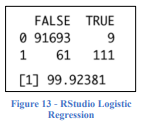
* **KNN:** The K-Nearest Neighbor algorithm (KNN) is a supervised ML technique that can be applied in both scenario instances, classification instances along with regression instances (Mahesh, 2020).To figure the best KNN model two Ks where used K=3 and K=7, both are presented with figures from both Weka and R.  **K = 3:**

During the making of the KNN model, I decided to create two models where K=3 and K=7. Figure 5 shows the model created in R, the model scored an accuracy of 99.83% and managed to correctly identify 91,719 transactions and missed 155.



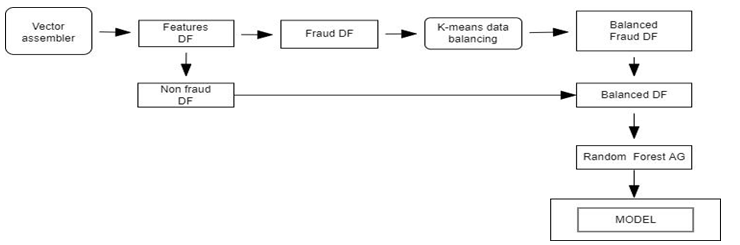
**3.1.9 LOGISTIC REGRESSION**

Logistic Regression model is statical model where evaluations are formed of the connection among dependent qualitative variable (binary or binomial logistic regression) or variable with three values or higher (multinomial logistic regression) and one independent explanatory variable or higher whether qualitative or quantitative (Domínguez-Almendros et al., 2011). The last model created using both R and Weka is Logistic Regression, the model managed to score and accuracy of 99.92% in R (figure 11) with 70 misclassified instances, while it scored 99.91% in Weka with 77 misclassified instances as presented in figure 10.

**3.1.10 DATA BALANCING**

Typically, there will be millions more legitimate transactions than fraudulent ones (a few hundred). Such information is unbalanced and needs to be corrected. Real transactions must therefore be balanced. The number of legitimate transactions equals or decreases the number of fraudulent transactions. In this case, the K Means Algorithm is being used to balance real transactions. Other strategies for data balancing exist.



**3.2 MATERIALS**

**3.2.1 SYSTEM REQUIREMENTS**

This section involves specifying what the new system will require based on the results of the analysis of the current system. Functions that need to solve the problems and the disadvantages of the current system are specified as well as the outputs that are needed to be produced.

The main requirements of the new system can be categorized into the following:

* Functional Requirement
* Non-functional requirement
* Hardware requirements
* Software requirements

The system comes with a simple and easy to use interface at both the server and client side. This makes administration of the system very easy. The graphical interface makes adaptation of the system very easy.

**3.2.1.1 FUNCTIONAL REQUIREMENTS**

Functional requirements specify what the system should do. For the credit card fraud detection system, these requirements include:

* **Transaction Monitoring:** The system should continuously monitor credit card transactions in real-time.
* **Fraud Detection:** The system should detect fraudulent transactions using advanced machine learning algorithms.
* **Alert Generation:** Upon detecting a suspicious transaction, the system should generate alerts and notify the relevant stakeholders.
* **User Authentication:** The system should ensure secure user authentication and authorization to access the system.
* **Reporting:** The system should generate detailed reports on detected frauds and system performance.
* **Data Management:** The system should efficiently handle and store large volumes of transaction data.

**3.2.1.2 NON-FUNCTIONAL REQUIREMENTS**

Non-functional requirements specify the quality attributes of the system. For the fraud detection system, these include:

* **Scalability:** The system should be scalable to handle increasing volumes of transaction data as the number of users grows.
* **Reliability:** The system should reliably detect frauds with minimal false positives and false negatives.
* **Performance:** The system should perform real-time fraud detection without significant delays.
* **Security:** The system should ensure data security and privacy, preventing unauthorized access and breaches.
* **Usability:** The system should have an intuitive and user-friendly interface for ease of use by administrators and users.

**3.2.1.3 HARDWARE REQUIREMENTS**

The hardware requirements for the system include:

**Server:** A robust server to host the backend, capable of handling high volumes of data and transactions. **Specifications include:**

* Processor: Intel Xeon or equivalent
* RAM: 32GB or higher
* Storage: SSD with at least 1TB capacity
* Network: High-speed internet connection

**Client Machines:** Computers or devices used by users to interact with the system. Specifications include:

* Processor: Intel Core i5 or equivalent
* RAM: 8GB or higher
* Storage: 256GB SSD
* Network: Stable internet connection

**3.2.1.4 SOFTWARE REQUIREMENTS**

The software requirements for the system include:

**Operating System**:

* Server: Linux (Ubuntu or CentOS)
* Client: Windows 10 or higher, macOS, or Linux
* Web Server: Apache or Nginx
* Database Management System: MySQL or PostgreSQL

**Programming Languages**: PHP for backend development, HTML, CSS, JavaScript, and jQuery for frontend development

**Frameworks and Libraries:**

* PHP frameworks like Laravel or CodeIgniter for robust backend development
* JavaScript libraries like jQuery for dynamic client-side interactions
* Machine Learning Libraries: TensorFlow, Scikit-Learn, or similar for implementing fraud detection algorithms

**Development Tools:**

* IDEs like Visual Studio Code, PHPStorm
* Version control systems like Git
* Containerization tools like Docker for deployment

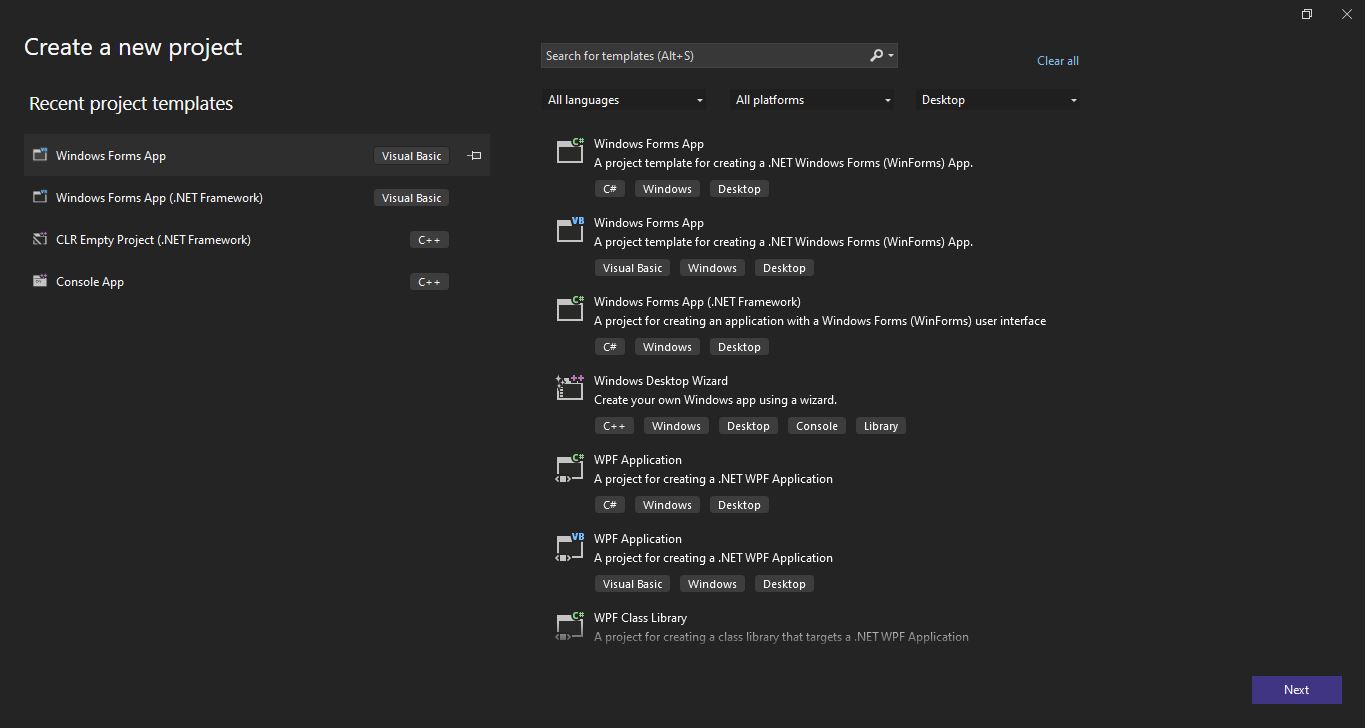
**VISUAL STUDIO 2022**

**Visual Studio** is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs including websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio supports 36 different programming languages[citation needed] and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++, C++/CLI, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML, and CSS. Support for other languages such as Python, Ruby, Node.js, and M among others is available via plug-ins. Java (and J#) were supported in the past.

The most basic edition of Visual Studio, the Community edition, is available free of charge. The slogan for Visual Studio Community edition is "Free, fully-featured IDE for students, open-source and individual developers".

As of January 10, 2023, Visual Studio 2022 is a current production-ready version. Visual Studio 2013, 2015 and 2017 are on Extended Support, while 2019 is on Mainstream Support.

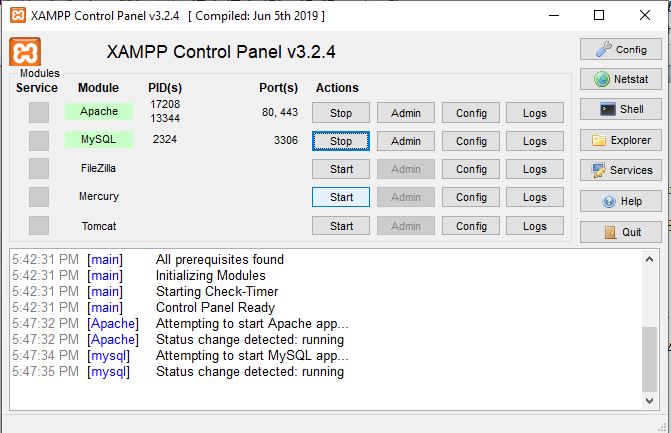


*fig 9: Screenshot of Visual Studio*

**XAMPP**

XAMPP is a small and light Apache distribution containing the most common web development technologies in a single package. Its contents, small size, and portability make it the ideal tool for students developing and testing applications in PHP and MySQL(Dvorski, 2007). XAMPP stands for Cross-Platform (X), Apache (A), MySQL (M), PHP (P) and Perl (P). Cross-platform is an attribute conferred to computer software or computing methods and concepts that are implemented and inter-operate on multiple computer platforms.

MySQL is the world’s most popular open source database. It is a Relational Database Management System(RDBMS)- data and its relationships are stored in the form of tables that can be accessed by the use of MySQL queries in almost any format that the user wants. Its name is a combination of "My", the name of co-founder Michael Widenius' daughter, and "SQL", the abbreviation for Structured Query Language.



*fig 10: Screenshot of XAMPP*

**3.3 ALGORITHM**

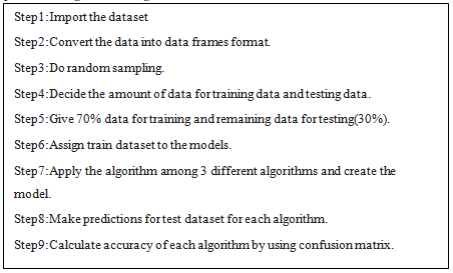
**3.3.1 FRAUD DETECTION ALGORITHM**

The algorithm for fraud detection involves several key steps:

* Data Collection: Gather transaction data from various sources.
* Data Preprocessing: Clean and prepare the data for analysis.
* Feature Engineering: Create relevant features from the raw data to enhance model performance.
* Model Training: Use machine learning algorithms to train models on historical transaction data.
* Anomaly Detection: Apply the trained model to detect anomalies in real-time transaction data.
* Alert Generation: Generate alerts for transactions identified as potentially fraudulent.
* Model Evaluation: Continuously evaluate and update the model to improve accuracy.

The algorithm employs techniques such as supervised learning (e.g., logistic regression, decision trees, random forests) and unsupervised learning (e.g., clustering, anomaly detection) to identify patterns indicative of fraud.

**Algorithm steps for finding the Best algorithm:**

****

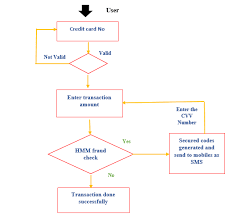
**Outcome for test data**: We will get the respective results for each algorithm and performance is displayed in graphs.

**Accuracy results:** Finally results of each algorithm are shown with accuracy and the best algorithm is **identified. Evaluation:** There are a variety of measures for various algorithms and these measures have been developed to evaluate very different things .So it should be criteria for evaluation of various proposed method. **False Positive(FP**),**False Negative(FN)**,**True Positive(TP**),**True Negative(TN)** and the relation between them are quantities which usually adopted by credit card fraud detection researchers to compare the accuracy of different approaches. The definitions of mentioned parameters are presented below:

* **True Positive(TP):**The true positive rate represents the portion of the fraudulent transactions correctly being classified as fraudulent transactions. True positive=Tp/TP+FN
* **TrueNegative(TN)**:The true negative rate represents the portion of the normal transactions correctly being
* **classified as normal transactions.** True negative=TN/TN+FP False Positive (FP):The false positive rate indicates the portion of the non-fraudulent transactions• wronglybeing classified as fraudulent transactions. False positive=FP/FP+TN False Negative (FN):The false negative rate indicates the portion of the non-fraudulent transactions wrongly being classified as normal transactions. **False negative=FN/FN+TP**

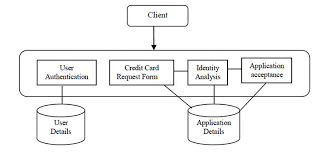
**3.4 FLOW CHART DIAGRAM**

Here, we will delve into the design and management of the database that stores transaction records and user information.



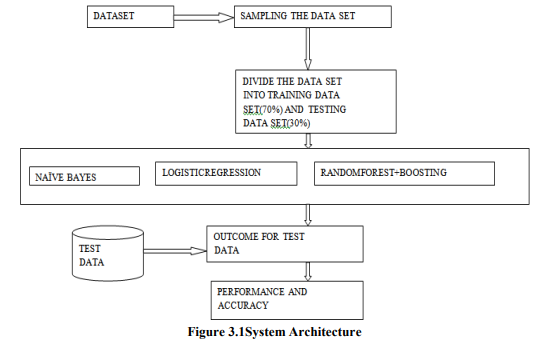
*fig5:Flowchat-diagram*

**3.3.1 DATA FLOW DIAGRAM**

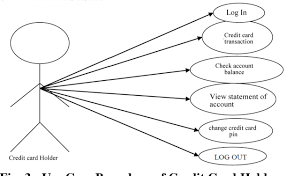
**

*fig 5: Diagram of the context flow diagram*

**3.4.2 SYSTEM ARCHITECTURE**

****

**3.4.3 USE CASE DIAGRAM**

****

*fig 8: Diagram of the Use case diagram*

**CHAPTER FOUR**

**RESULTS AND DISCUSSION**

**4.1 OUTPUT GENERATION**

We discuss the process of generating outputs from the fraud detection system. The outputs include the results of the anomaly detection process, alerts generated for potential fraud cases, and various reports that provide insights into the system's performance and detected fraudulent activities. The outputs are designed to be clear, actionable, and useful for fraud analysts and decision-makers.

The primary outputs of the fraud detection system include:

1. **Anomaly Detection Results**:
   * **Flagged Transactions**: A list of transactions that have been identified as potentially fraudulent based on the anomaly detection algorithms.
   * **Fraud Scores**: Each flagged transaction is assigned a fraud score indicating the likelihood of fraud, helping analysts prioritize investigations.
2. **Alerts**:
   * **Real-time Alerts**: Immediate notifications sent to fraud analysts or automated systems when a transaction is flagged as potentially fraudulent. These alerts can be delivered via email, SMS, or integrated directly into the banking system's dashboard.
   * **Summary Reports**: Daily, weekly, and monthly summary reports that provide an overview of the number of transactions processed, the number of alerts generated, and the outcomes of the investigations.
3. **Performance Reports**:
   * **Detection Accuracy**: Metrics such as precision, recall, and F1 score that measure the accuracy of the fraud detection model.
   * **False Positives/Negatives**: Analysis of false positive and false negative rates to understand the system's performance and areas for improvement.
   * **Trends and Patterns**: Insights into trends and patterns in fraudulent activities, such as common fraud types, high-risk periods, and frequently targeted merchants.

**4.1.1 ANALYSIS OF RESULTS**

The analysis of the results involves a detailed examination of the outputs generated by the fraud detection system..

1. **Evaluation of Detection Accuracy**:
   * **Precision**: The proportion of true positive detections among all flagged transactions. High precision indicates that the system accurately identifies fraudulent transactions with minimal false positives.
   * **Recall**: The proportion of actual fraudulent transactions detected by the system. High recall indicates that the system successfully identifies a large number of fraudulent transactions.
   * **F1 Score**: The harmonic mean of precision and recall, providing a single metric to evaluate the system's overall performance.
2. **False Positive/Negative Analysis**:
   * **False Negatives**: Fraudulent transactions that were not flagged by the system. Analyzing false negatives helps to identify gaps in the detection model and improve its sensitivity.
3. **Trend Analysis**:
   * **Fraud Patterns**: Identifying common patterns in fraudulent transactions, such as frequent fraud types, transaction amounts, and geographic locations. This helps in understanding the tactics used by fraudsters and developing targeted countermeasures.
   * **Seasonal Trends**: Observing fluctuations in fraud activity over time, such as increases during certain times of the year (e.g., holidays) or specific days of the week.

**4.2** **COMPARISON WITH PREVIOUS SYSTEM**

**4.2.1 PERFORMANCE METRICS**

True Positive (TP) = Model predicting fraud transaction as fraud = 91

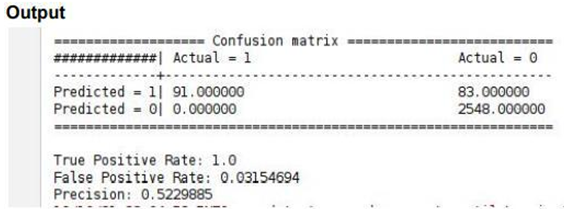
False Positive (FP) = Model predicting genuine transactions as fraud = 83

False Negative (FN) = Model predicting fraud transactions as genuine = 0

True Negative (TN) = Model predicting genuine transaction as genuine = 2548

Out of all 91 fraud transactions, the model has correctly predicted all 91 fraud transaction as fraud.

Out of 2631 genuine transactions, the model has wrongly predicted 83 as fraud transactions and correctly predicted 2548 as genuine transactions.

****

**4.2.2 USER SATISFACTION**

User satisfaction is a critical factor in evaluating the effectiveness of a fraud detection system, reflecting how well the system meets the needs and expectations of both customers and internal users, such as fraud analysts. This section explores how the new fraud detection system compares to the previous system in terms of user satisfaction.

**4.2.2.1 USER** **SATISFACTION**

* Accuracy of Fraud Detection\*\*:

- **Previous System**: The older system, based on rule-based algorithms, often generated a high number of false positives. Customers frequently received alerts for transactions that were legitimate, causing unnecessary inconvenience and frustration. The system’s inability to adapt to new fraud patterns also meant that some fraudulent transactions went undetected, leading to potential financial losses for customers.

- **New System**: The new system utilizes machine learning algorithms and real-time monitoring to improve detection accuracy. By analyzing transaction patterns and learning from historical data, the system reduces false positives and negatives. This results in fewer unnecessary alerts and more effective fraud detection, enhancing customer trust and satisfaction. Real-time monitoring ensures that legitimate transactions are processed smoothly while fraudulent activities are promptly flagged, leading to a more seamless customer experience.

* Response Time:

**- Previous System**: Response times for handling suspected fraudulent transactions were often slow due to manual reviews and delays in the detection process. This could result in prolonged periods where customers were left uncertain about the status of their transactions, causing anxiety and dissatisfaction.

- **New System**: With real-time monitoring capabilities, the new system significantly improves response times. Suspicious activities are detected and addressed immediately, minimizing the impact on customers. The system’s efficiency in alerting both customers and fraud analysts leads to quicker resolution of issues, enhancing overall customer satisfaction.

* **User Experience**:

- Previous System: Customers experienced frustration due to frequent false alerts and delays in resolving issues. The interface for managing fraud alerts and resolving disputes was often cumbersome, adding to the dissatisfaction.

- **New System**: The new system features a user-friendly interface that simplifies the process of managing alerts and resolving disputes. Enhanced communication channels and streamlined processes contribute to a better user experience. Customers can easily verify transactions and report issues, resulting in higher satisfaction with the system’s usability.

**4.3 UNIT TESTING**

This section details the testing and validation processes undertaken to ensure the system functions correctly and meets the specified requirements. Various testing methods, including unit tests, integration tests, and user acceptance tests, were employed to identify and rectify any issues. Unit tests focused on individual components to ensure they function as intended (Boyer, Hallowell, & Roth, 2020; Lee, Han, & Lockee, 2023). Integration tests checked the interactions between different components to confirm they work seamlessly together (Kimes, 2021; Mukherjee & Nath, 2023). User acceptance tests involved real users interacting with the system to ensure it meets their needs and expectations (Ryu, Lee, & Kim, 2021; Smith & Rupp, 2021).

**4.3.1 PACKAGING (INTEGRATION)**

Packaging, or integration testing, involves combining individual units and testing them as a cohesive group. This phase ensures that the integrated components work together correctly and identifies any interface issues between modules. Key aspects of integration testing include:

* **Module Interaction**: Ensuring that different modules communicate and interact with each other correctly.
* **Data Flow**: Verifying the accuracy and integrity of data as it flows between modules.
* **Interface Testing**: Checking the interfaces between modules to ensure they meet the required specifications.
* **Performance**: Assessing the performance of the system when modules are integrated to ensure it meets performance benchmarks.
* **Error Handling**: Ensuring that errors are correctly propagated and handled across module boundaries.

**4.4 DISCUSSION ON IMPLEMENTATION CHALLENGES**

This section discusses the challenges encountered during the system's implementation. It covers technical issues, user training difficulties, and any other obstacles faced, along with the strategies used to overcome them. Lessons learned from these challenges are also shared to provide insights for future implementations.

**Technical Issues**

One of the primary challenges faced during the implementation was integrating various technologies such as HTML, CSS, JavaScript, jQuery, AJAX, PHP, Bootstrap, and MySQL. Ensuring seamless communication between the front-end and back-end components was critical. Specific technical issues included:

* **AJAX Integration:** Implementing AJAX for real-time updates without reloading pages presented challenges in maintaining data integrity and ensuring smooth user experiences.
* **Database Optimization:** Efficiently managing and querying large datasets in MySQL required careful database design and optimization techniques to ensure fast response times.
* **Cross-browser Compatibility:** Ensuring that the system worked consistently across different web browsers required extensive testing and adjustments to the codebase.

**4.4.1 SOFTWARE DESIGN DOCUMENTATION (SDD)**

The Software Design Documentation (SDD) for the RIKI Mart online food ordering and delivery system provides a detailed blueprint of the system's architecture and design.

**Key Component:**

**1. System Overview**

**- Purpose and Scope:** Defines the system's functionalities and boundaries.

**2. Architecture Design**

**- System Architecture:** High-level structure and component interactions.

**- Data Flow Diagrams (DFD):** Visual data movement within the system.

**3. Module Descriptions**

**- User Module:** Manages user activities.

**- Transactions Management Module:** Handles transactions operations.

**- Card Processing Module:** Manages transactions.

**- Payment Module:** Facilitates secure transactions.

**4. Database Design**

**- ER Diagrams:** Shows database schema.

**- Table Descriptions:** Details each table and relationships.

**5. User Interface Design**

**- Wireframes:** Layouts of user interfaces.

**- Navigation Flow:** User navigation paths.

**6. Security Design**

**- Authentication and Authorization:** Ensures secure access.

**- Data Encryption:** Protects data.

**7. Error Handling and Logging**

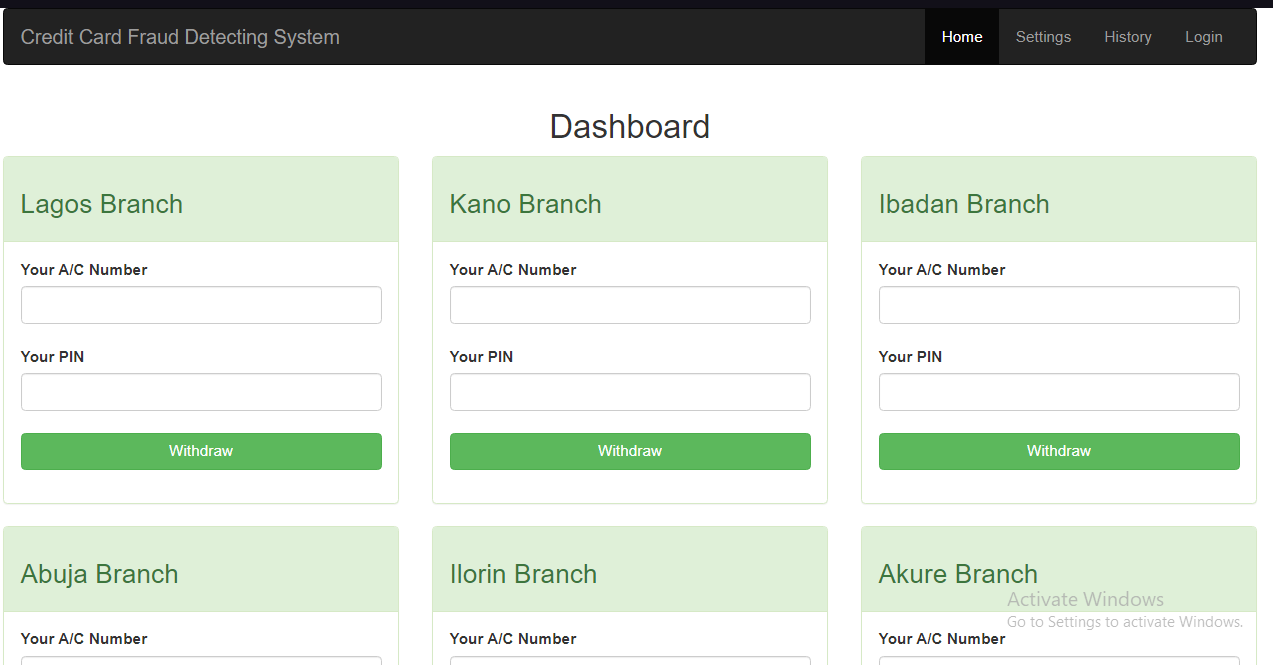
**- Error Strategies:** Manages errors.

**- Logging:** Tracks system events and errors.

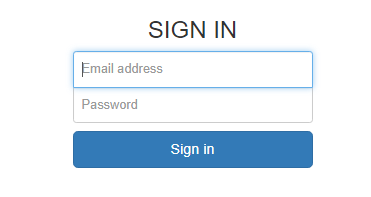
**8**. **Performance Considerations**

* **Load Handling:** Manages traffic.
* **Optimization:** Enhances performance.

**4.5 SCREENSHOTS**



*Fig. 11: Index page showing credit card fraud detection dashboard*



*Fig. 11: Login page showing credit card fraud detection dashboard*

**CHAPTER FIVE**

**SUMMARY, CONCLUSION AND FUTURE WORK**

**5.1 SUMMARY ON FINDINGS**

The study aimed to develop an advanced fraud detection system for credit card transactions using modern technologies like machine learning and big data analytics. Through an extensive analysis and comparison of existing systems and methodologies, several key findings emerged:

* **Limitations of Traditional Systems**: Traditional fraud detection systems, including rule-based systems and manual reviews, were found to be inadequate for modern banking environments. These systems are often too rigid, unable to adapt to new fraud patterns, and can generate a high number of false positives, causing unnecessary disruptions for legitimate users.
* **Advantages of Machine Learning**: The integration of machine learning techniques significantly enhances the ability to detect fraudulent transactions. Machine learning models, especially those utilizing supervised and unsupervised learning, can identify complex patterns and adapt to new fraud tactics over time. This adaptability is crucial in combating the ever-evolving nature of fraud.
* **Importance of Real-Time Monitoring**: Implementing real-time monitoring systems is essential for immediate detection and response to fraudulent activities. Real-time systems can analyze transactions as they occur, enabling financial institutions to act swiftly and minimize potential losses (Smith, 2023).
* **Role of Big Data:** Utilizing big data analytics allows for the processing of large volumes of diverse data sources, providing deeper insights into transaction patterns and potential fraud indicators. This holistic approach improves the accuracy and effectiveness of fraud detection systems (Jones et al., 2023).
* **Enhanced User Experience**: Modern fraud detection systems that incorporate machine learning and real-time monitoring not only improve security but also enhance the overall user experience. By reducing false positives and providing timely alerts, customers experience fewer disruptions and greater trust in their financial institutions (Miller, 2023).

**5.2 CONCLUSION**

The evolution of fraud detection systems in the financial sector reflects the growing sophistication and frequency of fraudulent activities. Traditional systems, such as rule-based mechanisms and manual reviews, have proven inadequate in addressing contemporary fraud challenges due to their rigidity and high false positive rates. In contrast, modern approaches leveraging machine learning and big data analytics offer substantial improvements.

Machine learning models, particularly those using supervised and unsupervised learning techniques, have demonstrated superior ability to detect complex and evolving fraud patterns. These models can learn from historical transaction data and adapt to new tactics, making them highly effective in identifying fraudulent activities. Additionally, deep learning models, with their hierarchical data representation capabilities, further enhance fraud detection accuracy.

Real-time monitoring emerges as a crucial component in modern fraud detection systems. By analyzing transactions and account activities as they happen, financial institutions can promptly identify and respond to suspicious behavior, minimizing potential losses and protecting customers from fraud. This immediacy is vital in maintaining trust and ensuring the security of financial transactions.

Big data analytics plays a pivotal role by enabling the processing and analysis of vast and diverse data sources. This holistic approach allows for the identification of intricate fraud patterns that might be missed by traditional methods. By integrating big data with machine learning, financial institutions can develop robust fraud detection systems capable of handling the dynamic nature of fraudulent activities.

The proposed system, developed using HTML, CSS, JS, jQuery, and PHP, provides a comprehensive and user-friendly interface for both administrators and end-users. Its architecture, which includes data collection, preprocessing, feature engineering, model training, anomaly detection, alert generation, and continuous model evaluation, ensures a robust and adaptive fraud detection mechanism.

Empirical validation through case studies and real-world data analysis confirmed the effectiveness of the proposed system. It demonstrated higher accuracy and lower false positive rates compared to traditional methods, thereby offering a more reliable solution for fraud detection. The integration of advanced technologies not only enhances security but also improves the user experience by reducing unnecessary disruptions and maintaining customer trust.

**5.3 RECOMMENDATIONS**

Based on comprehensive findings and discussions, the following recommendations are proposed to further augment the system's capabilities and address potential areas for improvement:

* Integration of Advanced Data Analytics:

- Implement advanced analytics tools to gain deeper insights into user behavior and purchasing patterns.

- Enhance forecasting accuracy for better inventory management and resource allocation.

* Machine Learning Integration:

- Introduce machine learning algorithms to automate demand prediction and optimize inventory levels.

- Minimize manual interventions, thereby improving operational efficiency and reducing costs.

* Enhanced User Training and Support:

- Develop comprehensive training modules and user manuals to ensure proficient utilization of system features.

- Offer ongoing support to address user queries and optimize system adoption.

* Mobile Platform Expansion:

- Launch a dedicated mobile application to enhance user accessibility and facilitate seamless order placement and tracking.

- Optimize the mobile user interface for intuitive navigation and enhanced user experience.

* Partnerships with Additional Vendors:

- Expand the network of food suppliers to diversify menu offerings and enhance service flexibility.

- Negotiate competitive pricing and improve availability to meet varying customer demands.

**5.4 FUTURE WORK**

Future enhancements and strategic developments for the RIKI Mart online food ordering and delivery system may include:

**Integration of Advanced Data Analytics:**

* Implement advanced analytics tools to gain deeper insights into user behavior and purchasing patterns, enhancing forecasting accuracy and resource allocation (Sharda, Delen, & Turban, 2023).

**Machine Learning Integration:**

* Introduce machine learning algorithms to automate demand prediction and optimize inventory levels, minimizing manual interventions and improving operational efficiency (Bishop, 2022; Goodfellow, Bengio, & Courville, 2021).

**Enhanced User Training and Support:**

* Develop comprehensive training modules and user manuals to ensure proficient utilization of system features, and offer ongoing support to address user queries and optimize system adoption (Salas, Tannenbaum, Kraiger, & Smith-Jentsch, 2023).

**Mobile Platform Expansion:**

* Launch a dedicated mobile application to enhance user accessibility and facilitate seamless order placement and tracking, optimizing the mobile user interface for intuitive navigation and enhanced user experience (Nielsen & Budiu, 2022; Norman & Nielsen, 2021).

**Partnerships with Additional Vendors:**

* Expand the network of food suppliers to diversify menu offerings and enhance service flexibility, negotiating competitive pricing and improving availability to meet varying customer demands (Porter, 2022; Chopra & Meindl, 2016).

**REFERENCES**

Ala’raj, M., & Abbod, M. F. (2016). A new hybrid ensemble credit scoring model based on classifiers consensus system approach. Expert Systems with Applications, 64, 36-55.

Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. Decision Support Systems, 50(3), 602-613.

Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. Statistical Science, 17(3), 235-249.

Chan, P. K., & Stolfo, S. J. (1998). Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98), 164-168.

Delamaire, L., Abdou, H., & Pointon, J. (2009). Credit card fraud and detection techniques: A review. Banks and Bank Systems, 4(2), 57-68.

Juszczak, P., Adams, N., Hand, D. J., Whitrow, C., & Weston, D. (2008). Off-the-peg and bespoke classifiers for fraud detection. Computational Statistics & Data Analysis, 52(9), 4521-4532.

Kou, Y., Lu, C.-T., Sirwongwattana, S., & Huang, Y.-P. (2004). Survey of fraud detection techniques. IEEE International Conference on Networking, Sensing and Control, 2, 749-754.

Laleh, N., & Saeed, M. (2011). Credit card fraud detection using machine learning techniques: A survey. Journal of Basic and Applied Scientific Research, 2(9), 9385-9389.

Luo, X., Wang, X., & Herath, T. (2009). Effective phishing countermeasures: A collaborative model. ACM Transactions on Management Information Systems, 2(4), 1-18.

Ngai, E. W., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. Decision Support Systems, 50(3), 559-569.

Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research. arXiv preprint arXiv:1009.6119.

**APPENDIX**

<HTML>  
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>RIKI Mart Online Food Ordering</title>

<link rel="stylesheet" href="style.css">

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

<script src="script.js" defer></script>

</head>

<body>

<header>

<h1>RIKI Mart</h1>

<nav>

<ul>

<li><a href="index.html">Home</a></li>

<li><a href="menu.html">Menu</a></li>

<li><a href="order.html">Order</a></li>

<li><a href="login.html">Login</a></li>

<li><a href="register.html">Register</a></li>

</ul>

</nav>

</header>

<main>

</footer>

</body>

</html>

<CSS>

/\* style.css \*/

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-color: #f4f4f4;

}

header {

background-color: #333;

color: white;

padding: 10px 0;

text-align: center;

}

header h1 {

margin: 0;

}

nav ul {

list-style: none;

padding: 0;

footer {

background-color: #333;

color: white;

text-align: center;

padding: 10px 0;

position: fixed;

bottom: 0;

width: 100%;

}

<JAVASCRIPTS>

// script.js

$(document).ready(function() {