

# **CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING ALGORITHMS**

**A PROJECT REPORT**

*Submitted by*

**ANA JESSICA K - 312319104014**

**FEBI V - 312319104031**

*in partial fulfillment of the award of the degree of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

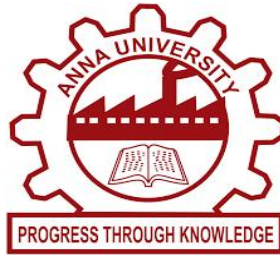


**St. JOSEPH'S COLLEGE OF ENGINEERING**  
**(An Autonomous Institution)**  
**OMR, Chennai 600 119**

**ANNA UNIVERSITY : CHENNAI 600 025**

**MARCH 2023**

## **ANNA UNIVERSITY : CHENNAI 600 025**



### **BONAFIDE CERTIFICATE**

Certified that this project report **“CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING ALGORITHMS”** is the bonafide work of **ANA JESSICA K (312319104014), FEBI V (312319104031)** who carried out the work under my guidance. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**SIGNATURE**

**HEAD OF THE DEPARTMENT**

Dr. A. Chandrasekar, M.E., Ph.D.,  
Professor & Head of Department  
Dept. of Computer Science and Engineering,  
St. Joseph's college of Engineering,  
OMR, Chennai-600 119

**SIGNATURE**

**SUPERVISOR**

Ms. S. Janani, M.E., (Ph.D.,)  
Assistant Professor,  
Dept. of Computer Science and Engineering,  
St. Joseph's college of Engineering,  
OMR Chennai-600 119

**Submitted to Project and Viva Examination held on \_\_\_\_\_**

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## ACKNOWLEDGEMENT

At the outset, we would like to express our sincere gratitude to our beloved **Dr. B. Babu Manoharan M.A., M.B.A., Ph.D., *Chairman, St. Joseph's Group of Institutions*** for his constant guidance and support to the student community and the Society.

I would like to express my hearty thanks to our respected ***Managing Director Mrs. S. Jessie Priya M.Com.*** for her kind encouragement and blessings.

I wish to express my sincere thanks to the ***Executive Director Mr. B. Shashi Sekar, M.Sc.*** for providing ample facilities in the institution.

I express sincere gratitude to our beloved **Principal Dr. Vaddi Seshagiri Rao M.E., M.B.A., Ph.D., F.I.E.** for his inspirational ideas during the course of the project.

I express my sincere gratitude to our beloved **Dean (Student Affairs) Dr. V. Vallinayagam M.Sc., M.Phil., Ph.D.,** and **Dean (Academics) Dr. G. Sreekumar M.Sc., M.Tech., Ph.D.,** for their inspirational ideas during the course of the project.

I wish to express our sincere thanks to **Dr. A. Chandrasekar M.E., Ph.D., *Head of the Department and Dean (Research)***, Department of Computer Science and Engineering, St. Joseph's College of Engineering for his guidance and assistance in solving the various intricacies involved in the project.

I would like to acknowledge my profound gratitude to our supervisor **Ms. S. Janani, M.E., (Ph.D.,)** for her expert guidance and connoisseur suggestion to carry out the study successfully.

Finally, I thank the **Faculty Members** and **my Family**, who helped and encouraged me constantly to complete the project successfully.

## **ABSTRACT**

Credit card fraud is an ever-growing problem in today's financial market. There has been a rapid increase in the rate of fraudulent activities in recent years causing a substantial financial loss to many organizations, companies, and government agencies. The numbers are expected to increase in the future, because of which, many researchers in this field have focused on detecting fraudulent behaviors early using advanced machine learning techniques. However, the credit card fraud detection is not an easy task because of two reasons: (i) the fraudulent behaviors usually differ for each attempt (ii) the dataset is highly imbalanced, i.e, the frequency of majority samples (genuine cases) outnumbers the minority samples (fraudulent cases). When providing input data of a highly unbalanced class distribution to the predictive model, the model tends to be biased towards the majority samples. As a result, it tends to misrepresent a fraudulent transaction as a genuine transaction. To tackle this problem different resampling methods along with ensemble models have been applied to the highly skewed dataset. Predictive models such as Logistic Regression, XGBoost and K-Nearest Neighbour in combination with different resampling techniques have been applied to predict if a transaction is fraudulent or genuine. The experimental results showed that the Stacking ensemble method in combination with a hybrid resampling approach of Synthetic Minority Over-sampling Technique (SMOTE) performed best.

# TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	<b>ABSTRACT</b>	iv
	<b>LIST OF FIGURES</b>	viii
	<b>LIST OF TABLES</b>	x
	<b>LIST OF ABBREVIATIONS</b>	xi
<b>1</b>	<b>INTRODUCTION</b>	
	1.1 Understanding the Background	1
	1.2 Working of Online Credit Card Transaction	3
	1.3 Scope of the Project	4
<b>2</b>	<b>SYSTEM ANALYSIS</b>	
	2.1 Existing System	6
	2.2 Drawbacks of Existing System	6
	2.3 Literature Survey	8
	2.4 Proposed System	11
	2.5 System Architecture	12
	2.5.1 Static Rules System	15
	2.5.2 Scoring Rules System	16
	2.5.3 Velocity Rules System	16
	2.5.4 Machine Learning Algorithms	17

<b>3</b>	<b>SYSTEM REQUIREMENTS</b>	
	3.1 Hardware Requirements	21
	3.2 Software Requirements	21
	3.3 System Packages Involved	21
<b>4</b>	<b>SYSTEM IMPLEMENTATION</b>	
	4.1 Dataset	25
	4.2 Pre-Processing the Data	27
	4.3 Data Analysis	33
	4.3.1 Univariate Analysis	33
	4.3.2 Correlation Plot	36
<b>5</b>	<b>SYSTEM DESIGN</b>	
	5.1 Data Flow Diagram	37
	5.1.1 Data Flow Diagram Level - 0	37
	5.1.2 Data Flow Diagram Level - 1	37
	5.1.3 Data Flow Diagram Level - 2	39
	5.2 Class Diagram	40
	5.3 Use Class Diagram	40
	5.4 Sequence Diagram	42

**RESULTS AND DISCUSSIONS**

6.1 Performance Analysis 43

6.2 Results and Discussions 48

6.3 Future Enhancements 49

**APPENDICES**

A. Screenshots of Demonstration 50

**REFERENCES** 52

## LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO
1.1	Classification of Financial Fraud	1
1.2	Statistic Diagram for Fraud Alert	2
2.1	Rules Based Systems	6
2.2	Estimation of Fraud Score	12
2.3	Credit Card Details Checkout Form	13
2.4	System Architecture of Freeze Fraud Framework	14
2.5	Static Rule Example	15
2.6	Scoring Rule Example	16
2.7	Averaging Method	18
2.8	Max Voting Method	19
2.9	Stacking Method	20
2.10	Bagging Method	20
4.1	Dataset Attributes	25
4.2	Absence of Null Values in the Dataset	28
4.3	Dataset with only Essential Columns	29
4.4	Dataset with risk labels	30
4.5	Dataset with Score values of each Modules	30
4.6	Methodology Representation	31
4.7	Representation of Imbalanced Dataset	32



4.8	Count Plot for address_verific Attributes	34
4.9	Count Plot for date_verific Attributes	34
4.10	Count Plot for email_verific Attributes	35
4.11	Count Plot for is_fraud Attributes	35
4.12	Heatmap of Correlated Variables	36
5.1	Data Flow Diagram Level - 0	37
5.2	Data Flow Diagram Level - 0	38
5.3	Data Flow Diagram Level - 0	39
5.4	Class Diagram	40
5.5	Use Class Diagram	41
5.6	Sequence Diagram	42
6.1	Comparison of accuracy of each ensemble model	46
6.2	Precision Recall Curve and Confusion Matrix of Stacking Ensemble Model	48

## LIST OF TABLES

TABLE NO.	TABLE NAME	PAGE NO.
4.1	Datatypes of the Attributes	26
6.1	Log - Loss of Ensemble Models	47

## **LIST OF ABBREVIATIONS**

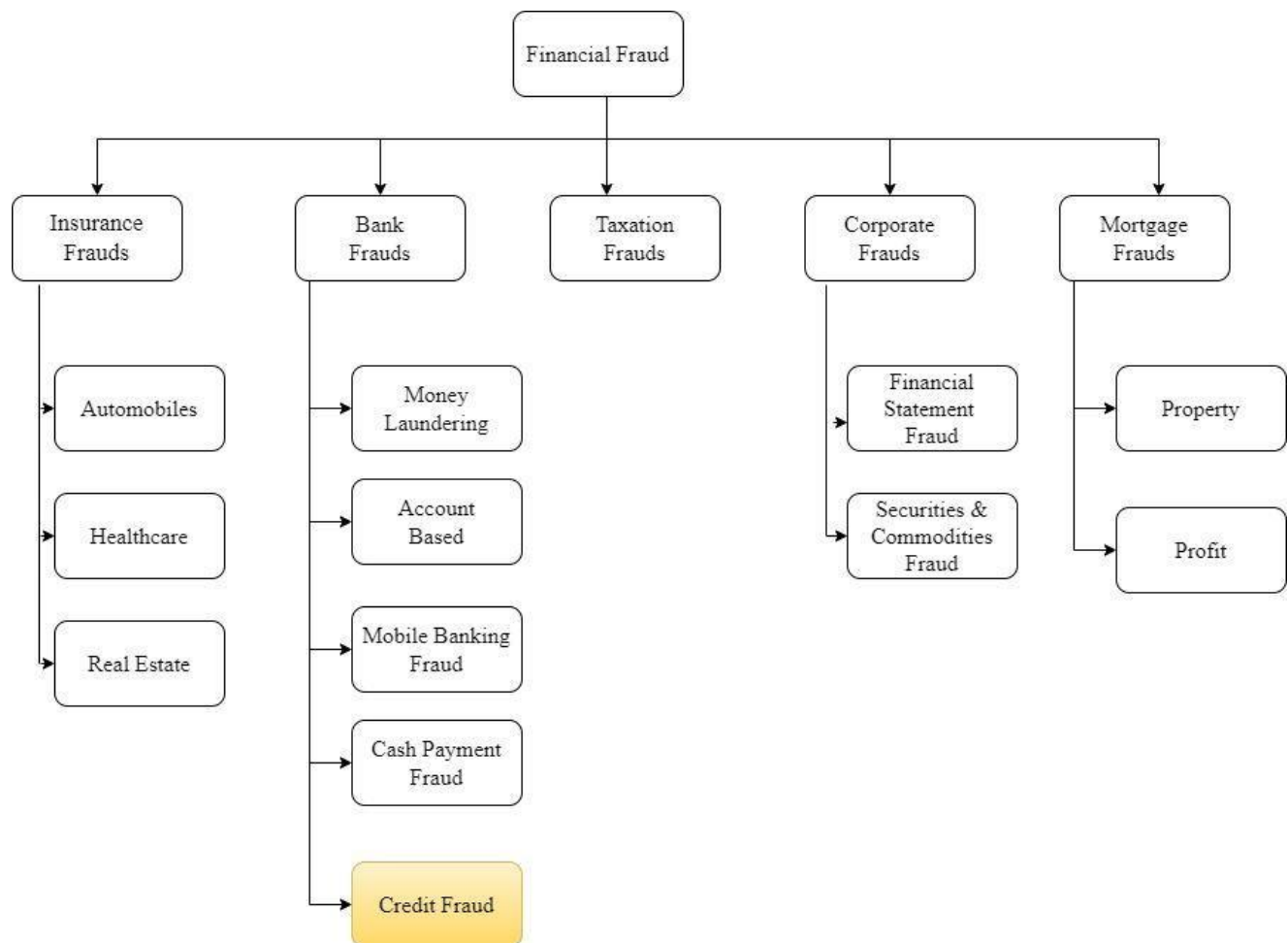
OTP	<b>One-Time Password</b>
CVV	<b>Card Verification Value</b>
CSIRT	<b>Computer Security Incident Response Team</b>
IP	<b>Internet Protocol</b>
XGBoost	<b>Extreme Gradient Boosting</b>
KNN	<b>K-Nearest Neighbors</b>
EDA	<b>Exploratory Data Analysis</b>
SMOTE	<b>Synthetic Minority Over-sampling Technique</b>
SMS	<b>Short Message Service</b>
NCRB	<b>National Crime Records Bureau</b>
AML	<b>Anti-Money Laundering</b>
CSV	<b>Comma-Separated Values</b>

# CHAPTER 1

## INTRODUCTION

### 1.1 UNDERSTANDING THE BACKGROUND

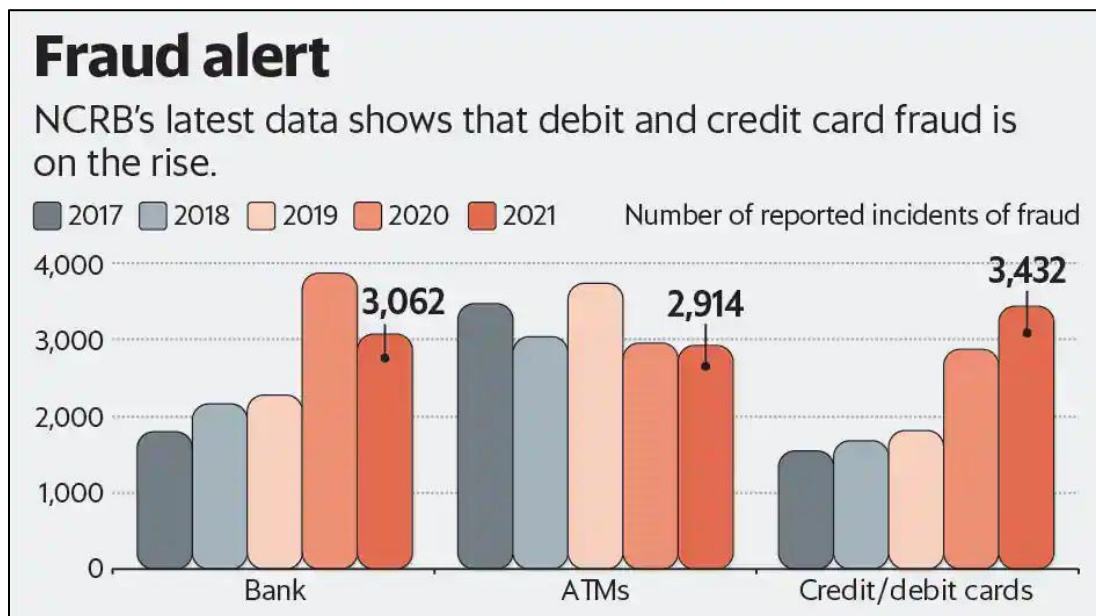
Financial fraud is a white-collar crime that affects the general public and has a negative impact on the whole economy. Often, these frauds involve misuse or manipulation of public funds by the fraudsters to make huge profits for themselves. With the advancement in the domain of technology, cases of financial fraud are on the rise. The cases of financial fraud as represented in figure 1.1, committed in cyberspace are no less daunting.



**Figure 1.1 Classification of Financial Fraud**

## Recent Statistics

According to the report released by the National Crime Records Bureau (NCRB) as shown in figure 1.2, 3432 cases of credit and debit card frauds were filed across India in 2021, an increase of nearly 20% from the earlier year. In 2020, such fraud cases increased by over 70% and in two years, credit and debit card-related frauds nearly doubled.



**Figure 1.2 Statistic Diagram for Fraud alert**

This project focuses on credit card fraud, where hackers can steal credit card information by using a phishing website or email, or obtain it from the Dark Web and misuse it.

## **1.2 WORKING OF ONLINE CREDIT CARD TRANSACTION**

Credit cards are now the most preferred way for customers to transact either offline or online. There are a number of reasons due to which consumers are slowly shifting from debit card transactions to credit cards, especially in developing countries like India.

Credit cards have become an increasingly popular form of payment in recent years, as they offer several advantages to users. Firstly, credit cards provide a convenient and secure way to make purchases, as users do not need to carry cash or worry about the safety of their funds. Secondly, credit cards often come with rewards programs, which can earn users cash back, travel points, or other incentives for using their card. Thirdly, credit cards can help build and improve credit scores, as responsible use and timely payments are reported to credit bureaus. Finally, credit cards can offer a variety of perks, such as purchase protection and extended warranties, that can provide added value to users. Overall, credit cards can be a valuable financial tool for those who use them responsibly and take advantage of their benefits.

When customers are paying for purchases online, the e-commerce store will ask for the following details:

- Choose whether it is Visa or MasterCard
- 16-digit card number
- Expiry date
- CVV
- Name as printed on the card
- Billing address

When they hit pay, the information is sent to the bank through a payment gateway. The bank sends an OTP or one-time password to the registered email ID or mobile number to authenticate the transaction. If the OTP is correct, then the transaction is complete.

However, with all these advantages, one downside that has been witnessed over the past few years of this increasing digital phenomenon is the rise of fraud on the credit card. Card-not-present fraud is committed when the criminal uses the details associated with the card such as the card number, account holder name and CVV code without having the card in their possession. In this project a solution, to detect and prevent ‘card-not-present fraud’ which is prevalent in recent digital transactions is provided.

### **1.3 SCOPE OF THE PROJECT**

Theft and fraud rates are increasing each year in online banking. In addition to the classic stealing of logins and passwords (phishing), Trojans on mobile phones can intercept OTP and SMS codes. Advanced Trojans (such as Zeus) can quickly replace an account entered by a customer with a thief’s account in the transaction system. Financial institutions are fully aware of this and invest capital in anti-fraud software and increase employment in dedicated threat response teams (CSIRTs) to ensure security and prevent hacker attacks. Various methods of threat detection are used by CSIRT. For example, a new type of Trojan appears that was written for a specific bank. The CSIRT investigates the fraud cases, and if the organization has a rule-based anti-fraud system, the team writes rules to block the threat. Otherwise, they can try to analyze Internet traffic even at the packet level to find anomalies and then add network rules on proxy or firewall servers that neutralize the threat.

The above detection methods work on a post-mortem debugging basis. Therefore, the actual fraud has already occurred, and the banks want to limit the scale of the attack so that other customers are not affected by this type of attack. Unfortunately, creating rules is not a solution to all problems. Some transactions fall into “gray” scoring; the bank cannot authorize every gray transaction, because it is impossible due to an insufficient number of employees in the call center. The bank’s business estimates the risk and bears its consequences in the form of financial losses.

In this project, a machine-learning based risk scoring system, is designed which introduces an early warning against fraud in the banking environment of any e-commerce site. The proposed system analyzes the client's transaction details and can find gray operations in the early stage before the client authentication and authorization process. This proposed method decreases the number of successful fraud occurrences.

The architecture and workflow of the project are innovative compared with existing anti-fraud systems for several reasons. The main advantage of this method is the flexibility of the architecture, which enables the solution to be quickly implemented in any organization. This is because costly and time-consuming modifications do not need to be introduced to the system’s functioning in the organization. The solutions currently used in the banking environment have a specific threat detection workflow, which limits their implementation possibilities. Most often, these are systems that are not prepared for trouble-free connection to the working infrastructure.



## CHAPTER 2

### SYSTEM ANALYSIS

#### 2.1 EXISTING SYSTEM

Fraud detection generally involves rules based techniques. These systems in figure 2.1, rely on hard coded rules that are set to flag transactions if they meet certain criteria. The rules are often expressed using “if-else” statements and are easily interpreted. They mirror the way in which a human would process a transaction. The engine checks if a transaction meets any of the risky patterns expressed in the rules and if it does, it blocks it or sends it to be manually reviewed by humans.

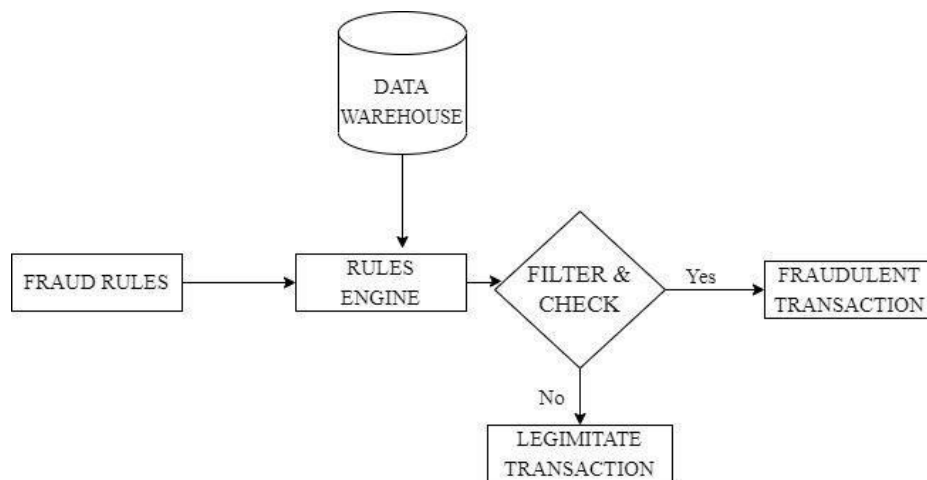


Figure 2.1 Rules Based System

#### 2.2 DRAWBACKS OF EXISTING SYSTEM

There are three critical weaknesses to rules-based systems. First, the rules engines do not scale. They must logically become nearly as complicated as the problem the system is trying to solve when rules must be added.

A 'rules explosion' occurs where a series of seemingly simple rules becomes a complex net of conflicting and overlapping rules. Thus they become very hard to understand. The more knowledge a user adds by adding more rules, the more complex and opaque the system becomes.

Second, these systems don't handle incomplete or incorrect information very well. Data that does not have an associated rule will be ignored. This means that rules-based systems are particularly bad at detecting 'unknowns' like, new derivations of malware or new disease epidemics.

Third, rules-based systems are unaware of variables that have an infinite number of possible values. Arbitrarily making these continuous variables into discrete variables may lead to missing patterns or deriving false patterns.

Hence, rules-based systems fall short in many ways:

- Rules-based systems approaches are time-intensive and cannot be used during real-time processing
- It requires manual work and supervision for detection of anomalies
- Rules-based requires multiple steps for verification that impede the user experience
- It can only identify obvious fraud patterns
- Continuous need of reverse engineering fraudsters' attacks - new rules have to be developed as new fraud patterns emerge.

## 2.3 LITERATURE SURVEY

Numerous pieces of literature about credit card fraud detection have been published already and are available for public usage.

Bora Mehar Sri Satya Teja et.al (2022)[1] discussed data collection and preprocessing, applying machine learning models, training, and testing the data were the modules covered in their paper. Exploration is done to overcome the problem of Concept drift to apply on real- world script. They have investigated the accuracy for only two classifiers, they are Decision Tree and Random Forest Classifier. The dataset is highly imbalanced to overcome that they have used Random Forest classifier model as the accuracy is 0.99963 which is higher than Decision tree and oversampling from SMOTE technique is done for better accuracy.

Emmanuel Ileberi et.al (2021)[2] discussed to alleviate the issue of class imbalance that is found in the European card dataset, this research investigated the use of the SMOTE. Moreover, the ML methods that were considered in this research include: Support Vector Machine (SVM), Random Forest (RF), Extra Tree (ET), Extreme Gradient Boosting (XGBoost), Logistic Regression (LR), and Decision Tree (DT). These ML methods were evaluated individually in terms of their effectiveness and classification quality. Additionally, the Adaptive Boosting (AdaBoost) algorithm was paired with each method to increase their robustness. In terms of the quality of classification, the ET-AdaBoost obtained an MCC of 0.99 and the XGB-AdaBoost achieved an MCC of 0.99.

Lakshmi S V S S and Selvani Deepthi Kavila (2018)[3] proposed the data set is highly imbalanced, it has about 0.172% of fraud transactions and the rest are genuine transactions. They have done oversampling to balance the data set, which resulted in 60% of fraud transactions and 40% genuine ones. Sensitivity, Specificity, accuracy and error rate are used to evaluate the performance for the proposed system. The accuracy for logistic regression, Decision tree and random forest classifier are 90.0, 94.3, and 95.5 respectively. By comparing all the three they have concluded that random forest has high accuracy.

Mosa M. M. Megdad et.al (2022)[4] paper proposed Exploratory Data Analysis (EDA) as a method for dataset analysis. As the dataset is imbalanced, they have split the dataset into three datasets: training, validating and testing. The ratio of the splitting was (60 x 20 x 20). They have trained and tested each model and recorded the results (accuracy, Precision, Recall, F1-score and time required for the training process in seconds). Same method was done after the dataset was balanced using SMOTE technique. The algorithms used in this study were: MLP Repressor, Random Forest Classifier, Complement NB, MLP Classifier, Gaussian NB, Bernoulli NB, LGBM Classifier, AdaBoost Classifier, K Neighbors Classifier, Logistic Regression, Bagging Classifier, Decision Tree Classifier and Deep Learning. They have concluded that there is no perfect model and there will always be a trade-off between precision and recall.

Shayan Wangde et.al (2022)[5] discussed triangulation that is how an online fraud transaction can be prevented. The three ways are Address Verification Service (AVS), Card Verification Value (CVV), Device Identification analysis.

Behavior and Location Analysis is used to reduce the number of positive false transactions identified as malicious by an FDS although they are genuine. EDA is done for the dataset comprising 9 feature columns and a goal column, as shown in the first rows below. Customer, ZipCodeOrigin, Merchant, ZipMerchant, Age, Gender, Category, Amount, Fraud are the feature columns. Five ML models are compared after the dataset is balanced using oversampling with SMOTE. They have also designed an ecommerce website which will block the transaction if the risk label is high considering the time module.

Fawaz Khaled Alarfaj et.al (2022)[6] discussed data collection and preprocessing, applying machine learning models, training, and testing the data were the modules covered in their papers. The detailed empirical analysis is carried out on the dataset, which improved the accuracy of detection of the frauds to some extent. Later, three architectures based on a convolutional neural network were applied to improve fraud detection performance. A comprehensive empirical analysis has been carried out by applying variations in the number of hidden layers, epochs and applying the latest models. Machine learning based approaches used for credit card detection are Extreme Learning Method, Decision Tree, Random Forest, Support Vector Machine, Logistic Regression and XG Boost. However, due to low accuracy, there is still a need to apply state of the art deep learning algorithms to reduce fraud losses. The evaluation of research work shows the improved results achieved, such as accuracy, f1-score, precision and AUC Curves. Comparing all the algorithm performances side to side, the CNN with 20 layers and the baseline model is the top method with an accuracy of 99.72%.

## **2.4 PROPOSED SYSYEM**

Rules-based systems tend to be more easily understood by human beings. If a user is confused by why a certain transaction of theirs was blocked, it is easier for a customer support associate to explain why a certain decision was made under a rules-based system. It can take advantage of knowledge shared by risk and fraud experts over years of experience and capture industry best practices. Rules can be iterated upon and improved over time, if the rule is fairly effective.

In recent years, machine learning systems have grown in popularity to complement rules-based systems. For big complex systems with large volumes of data, machine learning algorithms can be a powerful tool to find non-linear and non-intuitive patterns. The goal of the ML model is anomaly detection (discover / alerts on abnormal behavior).

This project provides the most robust solutions that require a mixture of both types of systems. One of the most common ways to use the strengths of both machine learning models and rules-based systems is to feed the output of the machine learning model into one of the inputs of the rules-based alerts, thus combining the best of both worlds.

By utilizing both machine learning and rules, it can capture fraudulent user behavior that is both human intuitive and complex as shown in figure 2.2. When it comes to online credit card fraud detection, this project uses both of these methodologies to provide the best solution.



**Figure 2.2 Estimation of Fraud Score**

In this project, a custom machine learning model is built to predict the outcome of each transaction in real-time and a unique fraud score for each transaction is found. The fraud or risk score ranges from 3-9 for every event or transaction. This score indicates the relative risk of fraud. With fraud score, admins are able to prioritize reviews of transactions based on risk. Based on the score, each event is segmented into one of 3 risk levels:

Low Risk (4-6): Low possibility of fraud, but may include false negatives (risk).

High Risk (7-9): High possibility of fraud, but may include false positives.

Legit (3): No possibility of fraud, but may include false negatives (risk).

## **2.5 SYSTEM ARCHITECTURE**

In e-commerce credit card transactions, card-not-present transactions are used through payment gateway as shown in figure 2.4. A payment gateway is a software application that communicates with the credit card company to obtain an authorization or a denial of a transaction.

The payment gateway encrypts the customer's information and forwards it to the credit card company. The credit card company checks that the consumer has the credit available, puts a hold on the funds and sends a message back that the sale has been approved.

### Review your plan


[Change plan](#)


**Premium Individual**  
1 Premium Account


Starting Today


One time payment of  
499.00 INR for 12 months


- [Offer terms](#) apply.

☐ Paytm Wallet  



☐ UPI  


☐ UPI (QR code)  



☐ Credit or debit card  


☒ Credit or debit card  


### Payment details




Card number

 0000 0000 0000 0000


Expiry date

MM / YY

Security code



Postal region

Choose a region 

By purchasing, you authorize Spotify to charge you the price above for the duration you selected. You agree that your right of withdrawal, including refund, is available within 14 days of purchase but is lost if you use Spotify during that time. If you have a recurring subscription, it will resume at the price in effect at the end of your prepaid period, unless you cancel earlier. Non-subscribers will return to a free account. No partial refunds. [Terms](#) apply. You consent to the tokenization of your card data by our authorized third-party token service provider, to enable a seamless checkout and future payments on our site.

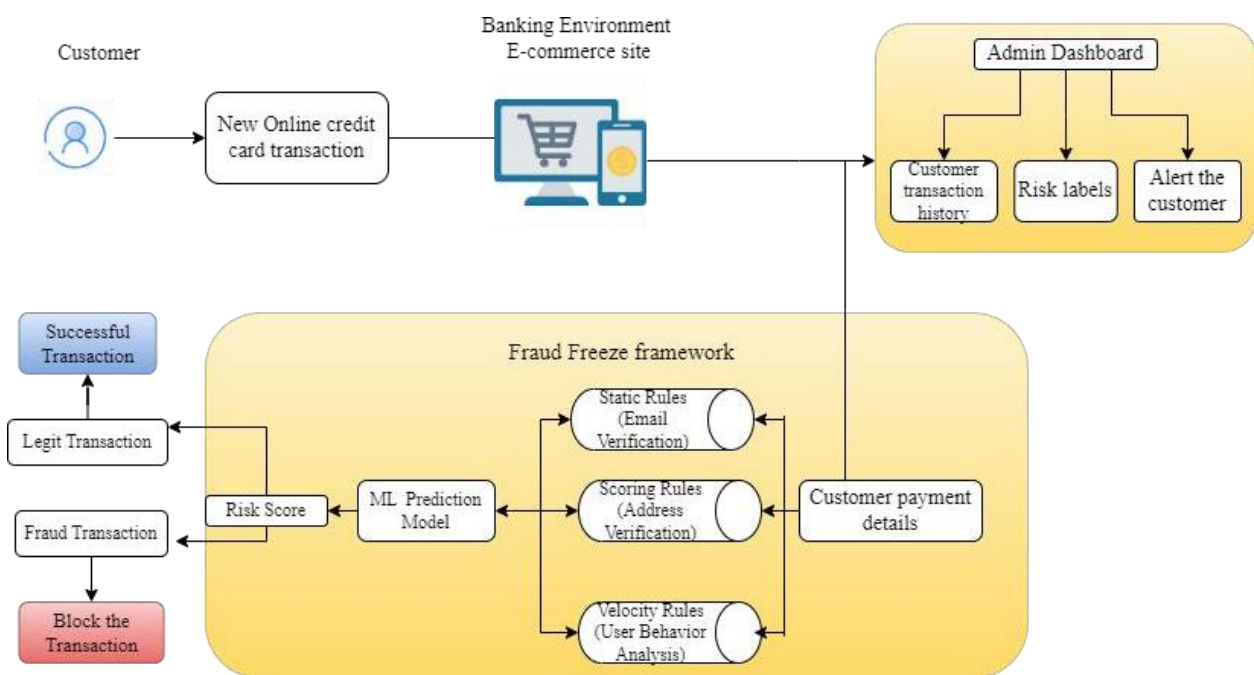
Figure 2.3 Credit Card Details Checkout form



Before the payment gateway process begins, “Fraud Freeze” framework shown in figure 2.4, will collect the customer payment details such as IP address of the customer, Address (City, State), E-mail Id, Transaction date and time

For privacy and security purposes, the customer’s credit card number, CCV (Card Verification Value) or Card Expiry Date is not stored or processed by this framework. With the collected necessary details, this framework processes the data into required format before it is analyzed by three modules. The three modules are:

- Static Rules System
- Scoring Rules System
- Velocity Rules System



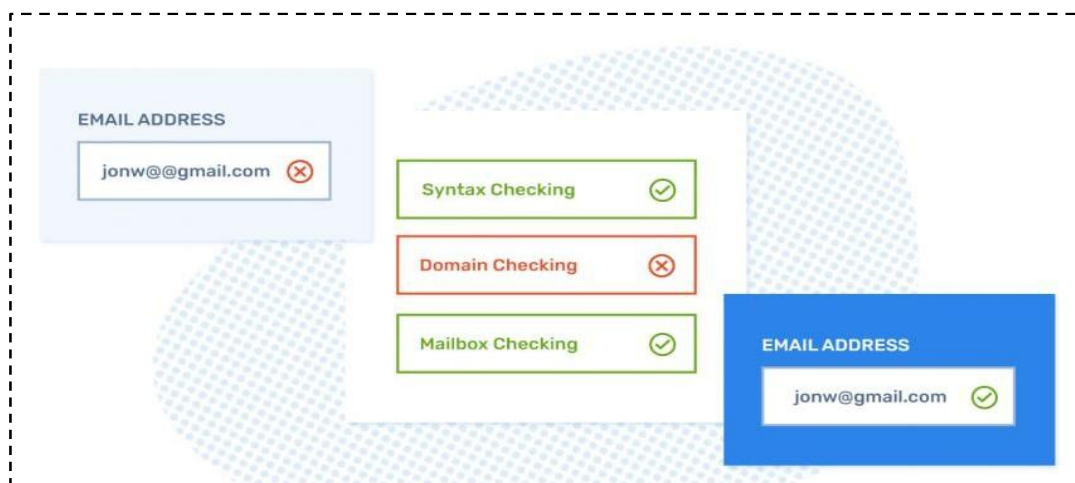
**Figure 2.4 System Architecture of Fraud Freeze Framework**

Each fraud detection rule helps to decide if an activity is fraudulent or not. It is based on correlation, statistics, and logical comparison. Each system is responsible for three different verification, whose results are fed into the custom trained machine learning model.

### 2.5.1 Static Rules System

A static rule is the most basic form of fraud rule and tends to follow a simple if/then logic. It is considered static when the outcome of the rule is strict and inflexible. Historically, the earliest static rules were like if an IP was found on a blacklist, the user would be blocked.

However, in this project static rules are creatively used based on the given dataset as shown in figure 2.5. It is verified if the provided customer's email-id is properly formatted and that its mailbox exists and is able to receive mail, without even sending an email. It also checks if the domain name is valid and whether it's a disposable email address or not.



**Figure 2.5 Static Rule Example**

### 2.5.2 Scoring Rules System

Fraud rules don't always have to block actions. This model has access to IP information and credit card details. So, address verification rule is made up of two parameters. The first parameter will hold the credit card details of a customer that point to the US. The second parameter will look at IP addresses of the same customer and it may point to Russia. Thus, based on the difference in the country code and the distance between IP address location and the billing address, a risk score is assigned as shown in figure 2.6.

CURRENT PARAMETERS

AND ▾

AND ▾

AND ▾

- Card country is equal to US
- IP country is equal to RU

Clear all

Delete group

Add group

Save rule Cancel

**Figure 2.6 Scoring Rule Example**

### 2.5.3 Velocity Rules System

Velocity rules attempt to understand the user behavior by looking at the customer transaction history over a time period.

A good example of a velocity rule would be when a customer whose transaction was previously blocked tries multiple times. If there are no barriers there, then they can use credential stuffing or brute force to crack credit card details.

Another example would be to monitor suspicious movements of money. This can be useful in the context of AML (anti money laundering). If there is an increase in spending (more than 200%) over a 24-hour period, it will trigger this kind of rule.

## **2.5.4 Machine Learning Algorithms**

All the three modules impact the model's fraud prediction score, and they are automatically generated as part of the fraud prediction. Each of them adds a score to ultimately calculate how risky the user action is and thus what the system should do about it.

Each fraud prediction also comes with a risk score between 3 and 9. Prediction explanations give the details of the influence of each event variable on the risk scores in terms of magnitude and direction labels. To identify top risk indicators during manual investigations each event is flagged with a risk label.

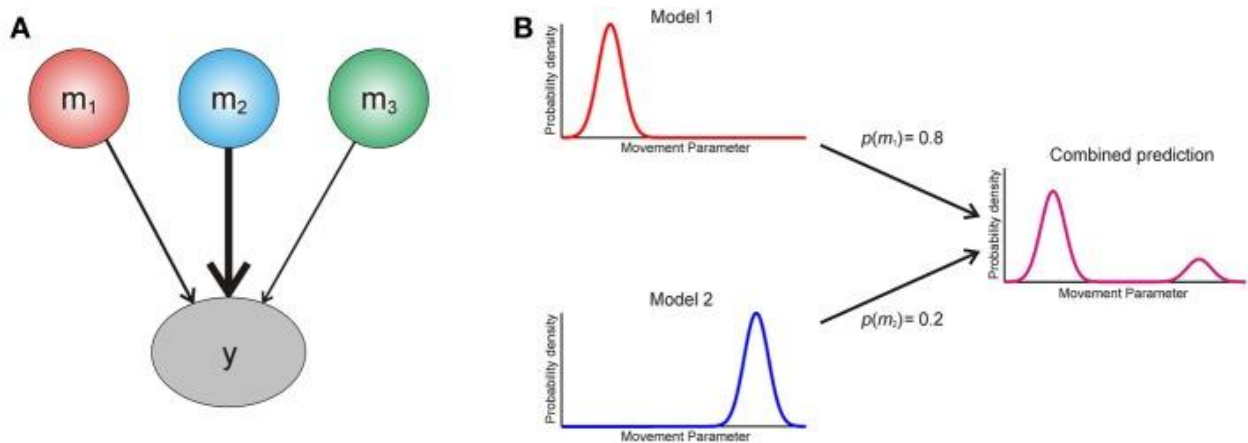
In the admin dashboard, green (3 points) means APPROVE, yellow (4-6 points) means REVIEW, and red (6+ points) means DECLINE. The system is setup so that all the rules add to a score of 9 and for a score of 3+, it automatically pauses the transaction and it is sent for manual review.

Thus, stacking multiple fraud rules allows to perform fraud scoring and the key of the "Fraud Freeze" framework is to adapt and customize the risk scores to each website's own business needs as the fraud scores aren't standardized. Thus, in this project fraud prevention is automated by setting thresholds based on the fraud scores and by using the pre-trained model's prediction.

To train the dataset Ensembling Models are used. Ensemble means a group of elements viewed as a wholerather than individually. An Ensemble method creates multiple models and combines them to solve it. Ensemble methods help to improve the robustness/generalizability of the model and boost its performance.

### Basic ensemble methods

- **Averaging method:** This method consists of building multiple models independently and returning the average of the prediction of all the models. In general, the combined output is better than an individual output because variance is reduced as shown in figure 2.7.



**Figure 2.7 Averaging Method**

- **Max Voting:** It is mainly used for classification problems. The method consists of building multiple models independently and getting their individual output called 'vote'. The class with maximum votes is returned as output. The classification models are combined using sklearn VotingClassifier and it is trained. The class with maximum votes is returned as output as shown in figure 2.8.

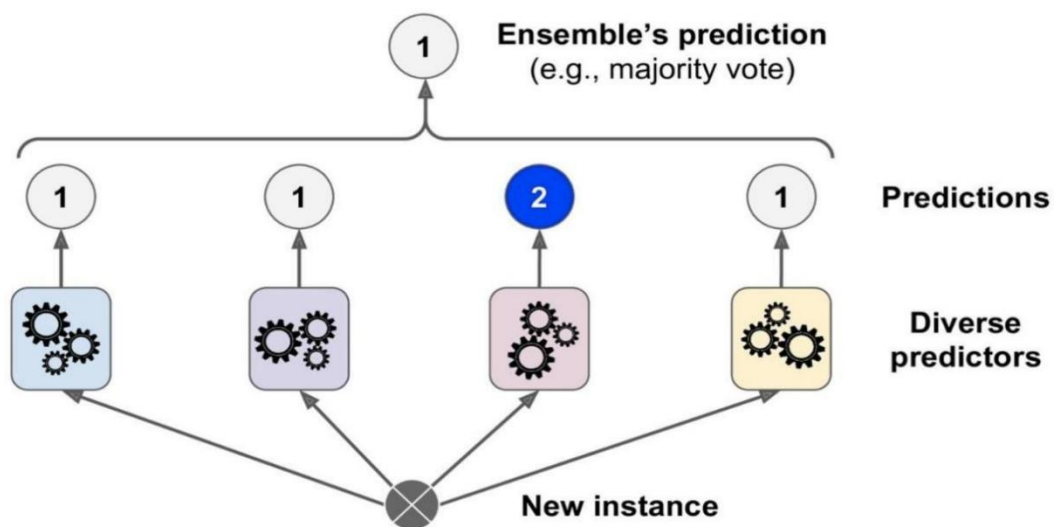
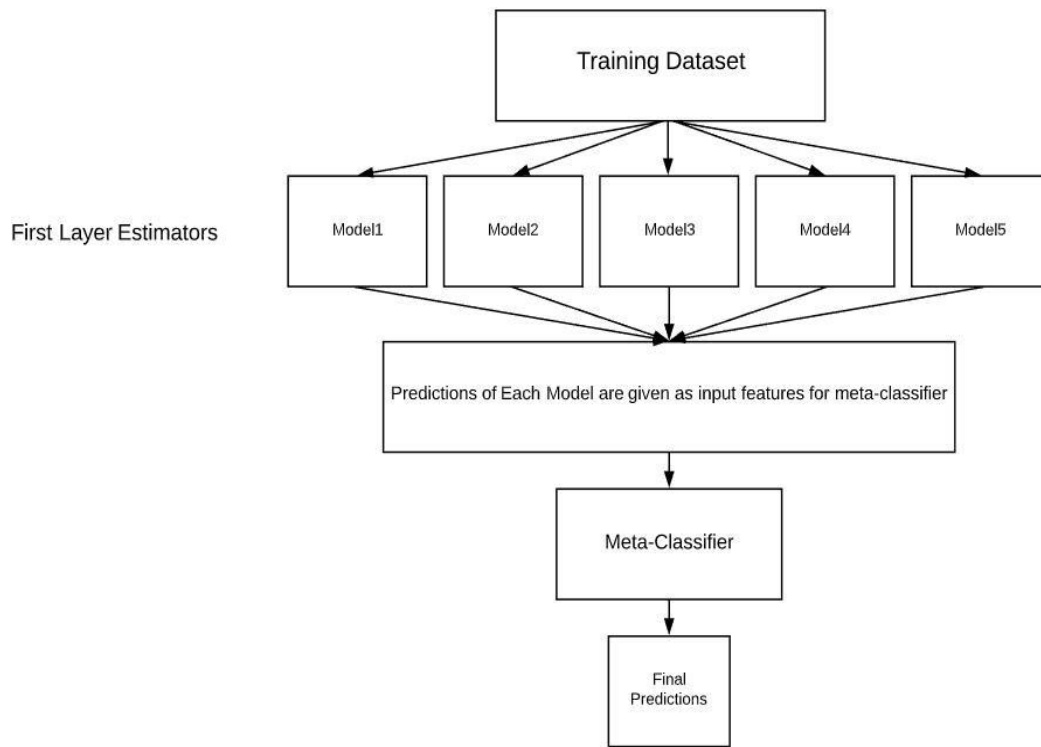


Figure 2.8 Max Voting Method

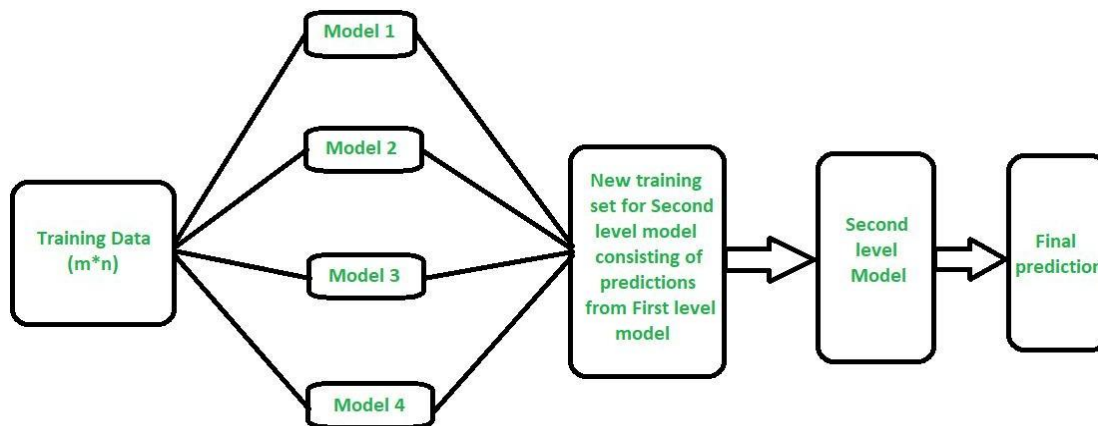
## Advanced Ensemble Methods

- **Stacking:** It is an ensemble method that combines multiple models via meta-model (meta-classifier or meta-regression). The base models are trained on the complete dataset, then the meta-model is trained on features returned (as output) from base models. The base models in stacking are typically different. The meta-model helps to find the features from base models to achieve the best accuracy as seen in figure 2.9.



**Figure 2.9 Stacking Method**

- **Bagging:** It is also known as a bootstrapping method. A bag is a subset of the dataset along with a replacement to make the size of the bag the same as the whole dataset. The final output is formed after combining the output of all base models as shown in figure 2.10.



**Figure 2.10 Bagging Method**

## **CHAPTER 3**

### **SYSTEM REQUIREMENTS**

#### **3.1 HARDWARE REQUIREMENTS**

- RAM: 8 GB and above
- Processor: 3 GHz and above
- Hard Disk: 5 GB and above
- CPU type: Intel Pentium 4
- Clock speed: 3.0 GH
- Monitor type: 17’’ Colored monitor
- Keyboard type: 122 keys
- CD -drive type: 52x max

#### **3.2 SOFTWARE REQUIREMENTS**

- Operating system: Windows, Linux, MacOS
- Scripting Language: Python
- Technology: Machine Learning
- IDE: VSCode

#### **3.3 SYSTEM PACKAGE INVOLVED**

The models are implemented using Python 3.7 with listed libraries:



## Imblearn

Imbalanced-learn (imported as `imblearn`) is an open source, MIT-licensed library relying on `scikit-learn` (imported as `sklearn`) and provides tools when dealing with classification with imbalanced classes. This module helps in balancing the datasets which are highly skewed or biased towards some classes. Thus, it helps in resampling the classes which are otherwise oversampled or undersampled. If there is a greater imbalance ratio, the output is biased to the class which has a higher number of examples.

## Sklearn

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 consistent interface in Python.

- The **`sklearn.metrics`** module implements several loss, score, and utility functions to measure classification performance. Some metrics might require probability estimates of the positive class, confidence values, or binary decisions values.
- The **`sklearn.preprocessing`** package provides several common utility functions and transformer classes to change raw feature vectors into a representation that is more suitable for the downstream estimators. In general, learning algorithms benefit from standardization of the dataset.

- **ensemble** learning uses multiple machine learning models to try to make better predictions on a dataset. An ensemble model works by training different models on a dataset and having each model make predictions individually. The predictions of these models are then combined in the ensemble model to make a final prediction.
- **linear\_model** is a class of the sklearn module that contains different functions for performing machine learning with linear models. The term linear model implies that the model is specified as a linear combination of features.
- **sklearn.neighbors** provides functionality for unsupervised and supervised neighbors-based learning methods. Unsupervised nearest neighbors is the foundation of many other learning methods, notably manifold learning and spectral clustering.
- **Naive Bayes classifier** is one of the simplest supervised learning algorithms. This classifier is based on Bayes Theorem. It is a fast, accurate and reliable algorithm. Naive Bayes classifiers have high accuracy and speed on large datasets. Naive Bayes classifier assumes that the effect of a particular feature in a class is independent of other features.

## **Flask**

Flask is a python web framework, that is used to develop web applications easily. It has a small and easy-to-extend core. It is a microframework that does not include an ORM (Object Relational Manager) or such features. It is based on the WSGI toolkit and the Jinja2 template engine.

## **Geopy**

Geopy is a Python client for several popular geocoding web services. Geopy makes it easy for Python developers to locate the coordinates of addresses, cities, countries, and landmarks across the globe using third-party geocoders and other data sources. geopy includes geocoder classes for the OpenStreetMap Nominatim, Google Geocoding API (V3), and many other geocoding services. The full list is available on the Geocoders doc section. Geocoder classes are located in `geopy.geocoders`.

## **Email\_validator**

It is a robust Python library that validates email addresses. It performs two types of validation - syntax validation and deliverability validation. That is important because the email address must meet the required form and have a resolvable domain name at the same time to be considered valid.

# CHAPTER 4

## SYSTEM IMPLEMENTATION

### 4.1 DATASET

The real-time sample dataset used in this project provides details of online ‘Transaction Fraud Insights’. This public dataset is provided by ‘aws-fraud-detector-samples’ github repository. The dataset is in a text file that uses comma-separated value (CSV) in the UTF-8 format. The first row of the CSV dataset file contains the headers. Each of these rows consists of data elements from a single account transaction.

The file transaction\_data\_100K\_full.csv contains variables for each event and they include LABEL\_TIMESTAMP, EVENT\_ID, ENTITY\_TYPE, ENTITY\_ID, card\_bin, customer\_name, billing\_street, billing\_city, billing\_country, customer\_job, ip\_address, customer\_email, billing\_phone, user\_agent, product\_category, order\_price, payment\_currency and merchant as shown in figure 4.1. The data file also contains two mandatory fields:

- EVENT\_TIMESTAMP – Defines when the event occurred
- EVENT\_LABEL – Classifies the event as fraudulent or legitimate

Event metadata			Event variables					Event dataset
EVENT_TIMESTAMP,	EVENT_ID,	EVENT_LABEL,	email_address,	phone_number,	billing_street,	billing_state,	ip_address	
2020-12-06T03:13:34Z,	R12345,	fraud,	regular1@example.com,	110-345-0990,	mayhem ave,	OH,	112.136.132.151	
2020-11-13T12:47:00Z,	P56890,	legit,	premium1@example.com,	112-890-4532,	howie lane,	KY,	192.169.234.143	
2021-02-19T22:52:43Z,	R10001,	legit,	regular2@example.net,	078-777-5555,	lankhurst dr,	HI,	185.112.224.79	
2020-11-29T00:16:09Z,	R56099,	fraud,	regular3@example.edu,	777-213-0033,	noland ave,	IL,	68.73.183.186	
2021-01-16T07:30:03Z,	P08954,	legit,	premium2@example.net,	444-040-8344,	oakwood apt,	MA,	117.65.246.206	

Figure 4.1 Dataset Attributes

**Event metadata-** provides information about the event. For example, EVENT\_TIMESTAMP is an event metadata that specifies the time event occurred.

**Event variable-** represents the data elements that are specific to each event which we want to use for creating and training the fraud detection model.

**Event data-** represents the data collected from the actual event. In the CSV file, each row contains data from a single transaction. Each data element in the row matches with the corresponding event metadata or the event variable. The data types of the attributes are in table 4.1.

**Table 4.1 Datatypes of the Attributes**

S.No	Columns	Data Type
1	EVENT_LABEL	int64
2	EVENT_TIMESTAMP	object
3	LABEL_TIMESTAMP	object
4	EVENT_ID	object
5	ENTITY_TYPE	object
6	ENTITY_ID	object
7	card_bin	int64
8	customer_name	object
9	billing_street	object
10	billing_city	object

11	billing_state	object
12	billing_zip	int64
13	billing_latitude	float64
14	billing_longitude	float64
15	billing_country	object
16	customer_job	object
17	ip_address	object
18	customer_email	object
19	billing_phone	object
20	user_agent	object
21	product_category	object
22	order_price	float64
23	payment_currency	object
24	merchant	object

## 4.2 PRE-PROCESSING THE DATA

Null values are a big problem in machine learning. Since sklearn is used packages, it is required to clean up null values before the data is passed to the machine learning or deep learning model. Hence, the null values are checked and handled.

```

[6]: df.isnull().sum()

[6]: EVENT_LABEL      0
      EVENT_TIMESTAMP  0
      LABEL_TIMESTAMP  0
      EVENT_ID        0
      ENTITY_TYPE     0
      ENTITY_ID       0
      card_bin        0
      customer_name    0
      billing_street    0
      billing_city     0
      billing_state    0
      billing_zip      0
      billing_latitude  0
      billing_longitude 0
      billing_country  0
      customer_job     0
      ip_address       0
      customer_email   0
      billing_phone    0
      user_agent       0
      product_category 0
      order_price      0
      payment_currency 0
      merchant         0
      dtype: int64

```

**Figure 4.2 Absence of Null Values in the Dataset**

Then unnecessary columns in the dataset like LABEL\_TIMESTAMP, EVENT\_ID, ENTITY\_TYPE, ENTITY\_ID, product\_category, customer\_name, order\_price, card\_bin, merchant, customer\_job, billing\_phone, user\_agent, merchant, payment\_currency, billing\_street, billing\_city, 'billing\_state, billing\_zip are removed and EVENT\_TIMESTAMP as transac\_time and EVENT\_LABEL as is\_fraud were renamed. Then the transac\_time column is split into date and time separately.

Thus are modified dataset looks like figure 4.3.



```
[14]: df.head()
```

	is_fraud	billing_latitude	billing_longitude	billing_country	ip_address	customer_email	time	date
0	0	37.6092	-84.4269	US	120.79.45.214	woodardbrenda@gmail.com	09:13:44	2022-08-12
1	0	36.4873	-79.4041	US	212.42.56.229	scottdalton@robinson.biz	08:16:25	2022-07-11
2	0	31.4420	-84.7241	US	164.238.228.201	pjohnson@cruz.info	10:58:42	2022-08-18
3	0	32.7783	-117.1335	US	210.108.230.215	progers@hansen-yu.com	15:52:36	2022-06-10
4	0	43.1411	-74.2444	US	189.103.115.129	reevesmichael@gmail.com	16:35:00	2022-08-30

**Figure 4.3 Dataset with only Essential Columns**

The **static rule system** takes in customer email as its input and checks using the ‘email\_validator’ python package and it gives a high risk label if it is found to be a fraud email-id or else legit label.

The **scoring rule system** takes in the customer's IP address as its input and derives the customer’s current location coordinates and country code with the billing address entered by the customer. Based on the difference between those parameters, high risk, low risk and legit labels are allocated.

The **velocity rule system** monitors the transaction history and does behavior analysis of the customer and allocates high risk, low risk and legit labels

Thus the risk values are estimated using the three modules and three new columns are derived by applying static rule, scoring rule and velocity rule, shown in figure 4.4.



[42]:

```
df_new.head()
```

[42...]

	is_fraud	address_verific	date_verific	email_verific
0	0	high risk	legit	legit
1	0	legit	low risk	legit
2	0	low risk	legit	legit
3	0	high risk	legit	legit
4	0	legit	legit	legit

**Figure 4.4 Dataset with risk labels**

The data is converted into numerical values to train the machine learning model and total risk score is also found. Thus, the final dataset appears to be, as shown in figure 4.5.

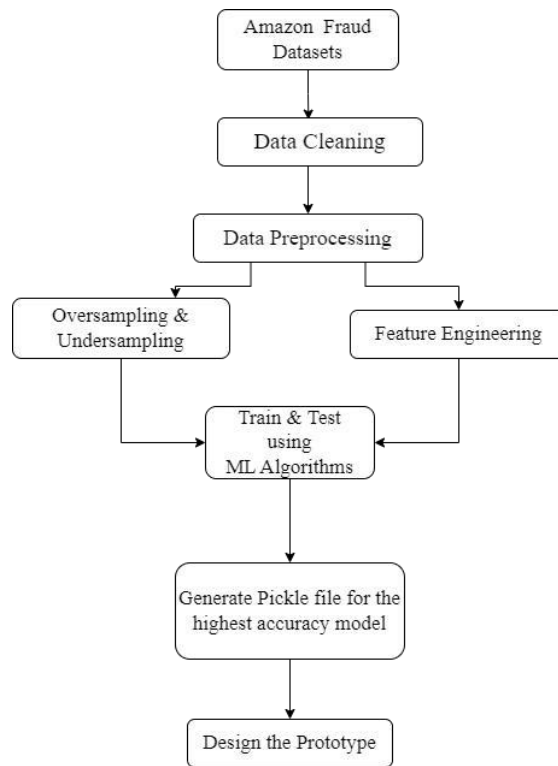
[46]:

```
df_new.head(10)
```

[46...]

	is_fraud	address_verific	date_verific	email_verific	risk_score
0	0	3	1	1	5
1	0	1	2	1	4
2	0	2	1	1	4
3	0	3	1	1	5
4	0	1	1	1	3
5	0	1	2	1	4
6	1	3	2	3	8
7	0	1	1	1	3
8	0	3	2	1	6
9	0	1	1	1	3

**Figure 4.5 Dataset with Score Values of each Module**



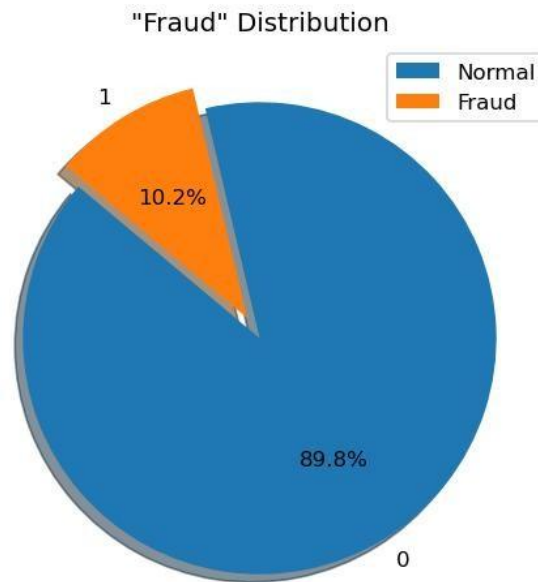
**Figure 4.6 Methodology Representation**

After preprocessing the data, figure 4.6 shows the methodology that is followed. The dataset is then divided into training and testing dataset in 80-20 fashion. The size of each set is as follows.

```
Train Size = 93364
Test Size = 23341
Total Size = 116705
```

This dataset is highly imbalanced as seen in figure 4.7. The problem with imbalanced classification is that there are too few examples of the minority class for a model to effectively learn the decision boundary.

So the examples in the minority class must be oversampled. This can be achieved by simply duplicating examples from the minority class in the training dataset prior to fitting a model. This can balance the class distribution but does not provide any additional information to the model.



**Figure 4.7 Representation of Imbalanced Dataset**

The most widely used approach to synthesizing new examples is called the Synthetic Minority Oversampling Technique, or SMOTE for short. As seen below, the count of both majority class and minority class are the same after applying SMOTE technique.

```
Before OverSampling, counts of label '1': 9642
Before OverSampling, counts of label '0': 83722

After OverSampling, the shape of train_X: (167444, 5)
After OverSampling, the shape of train_y: (167444,)

After OverSampling, counts of label '1': 83722
After OverSampling, counts of label '0': 83722
```

To train the oversampled dataset machine learning algorithm based on esembling methods are used.

## 4.3 DATA ANALYSIS

Exploratory Data Analysis (EDA) was performed to gain a better understanding of data aspects like,

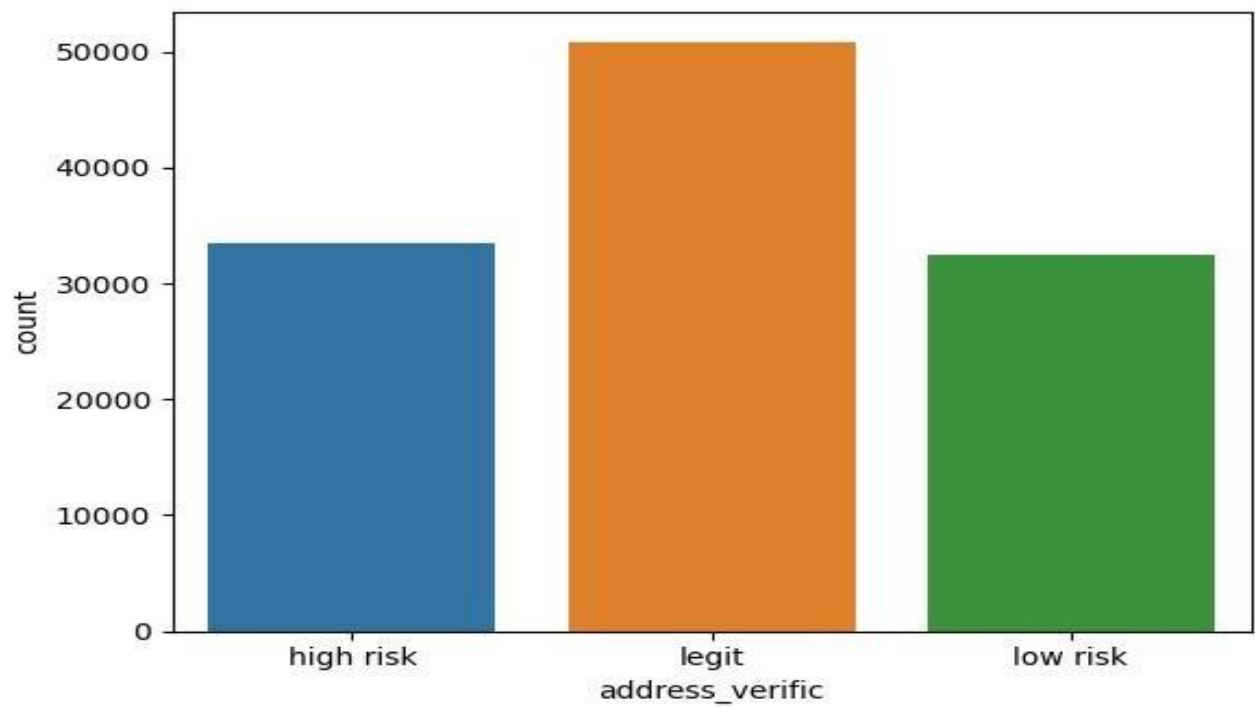
- main features of data
- variables and relationships that hold between them
- identifying which variables are important for our problem

### 4.3.1 Univariate Analysis

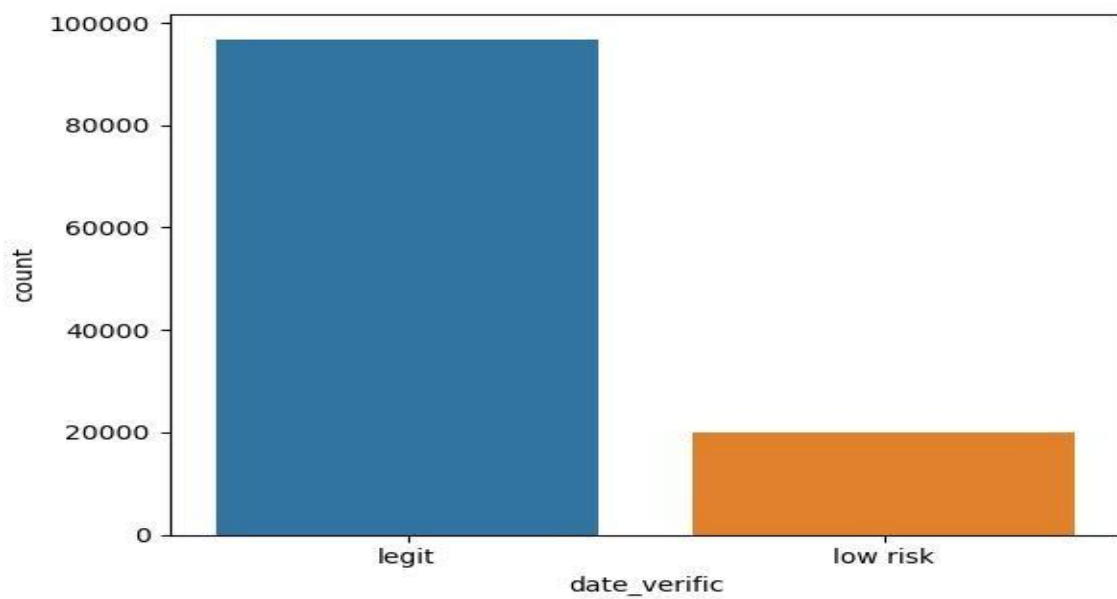
It is the easiest form of analyzing data where each variable is analyzed individually. For categorical features, count plots are used to calculate the number of each category in a particular variable.

#### Categorical Variables

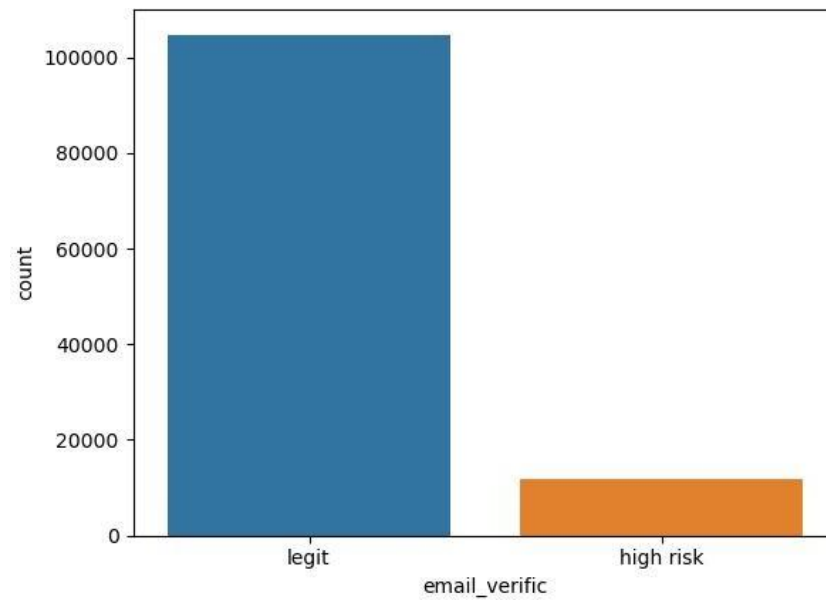
A count plot compares different classes of a categorical feature and how often they occur. It has been inferred from below count plot that `address_verific` attribute is the most important attribute to find high risk transactions. `date_verific` attribute is the least important attribute as it has no high risk label as shown in figure 4.9. `email_verific` attribute has no low risk label as inferred from below figure 4.10.



**Figure 4.8 Count plot for address\_verific Attribute**



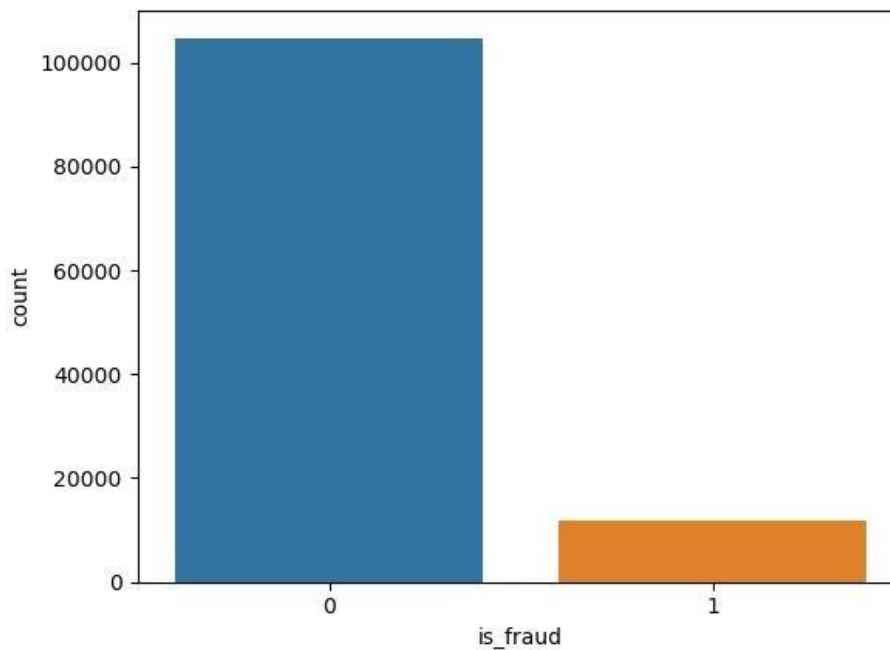
**Figure 4.9 Count plot for date\_verific Attribute**



**Figure 4.10 Count plot for email\_verific Attribute**

### Numerical Variable

For numerical variable also count plot has been used to observe the distribution of the variable.



**Figure 4.11 Count plot for is\_fraud Attribute**

It can be inferred from the figure 4.11 that this dataset has less number of fraud transactions than legit transactions.

#### 4.4.2 Correlation Plot

To understand the correlation between all the numerical variables, the heat map is used and it helps to visualize the correlation. It visualize data through variations in coloring as shown in 4.12. The variables, with darker colors means their correlation is more.

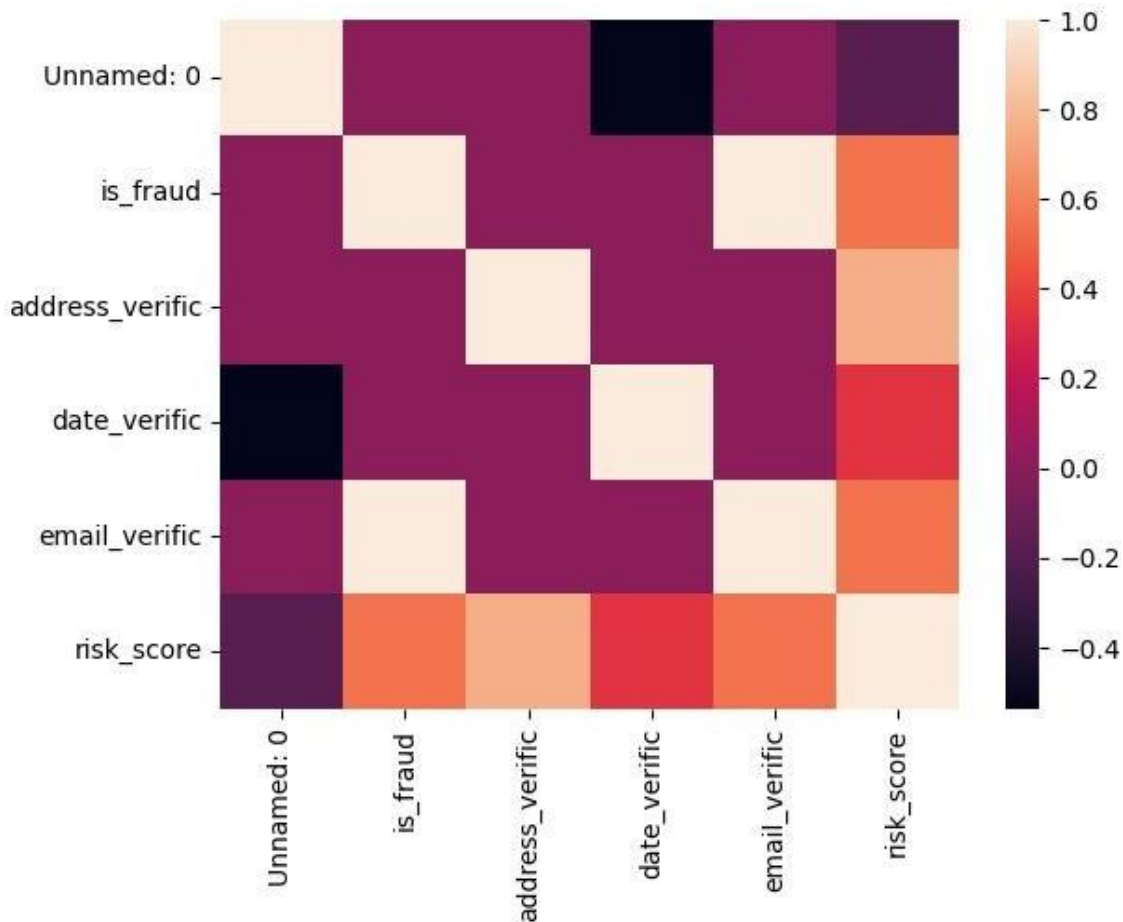


Figure 4.12 Heatmap of the Correlated Variabl

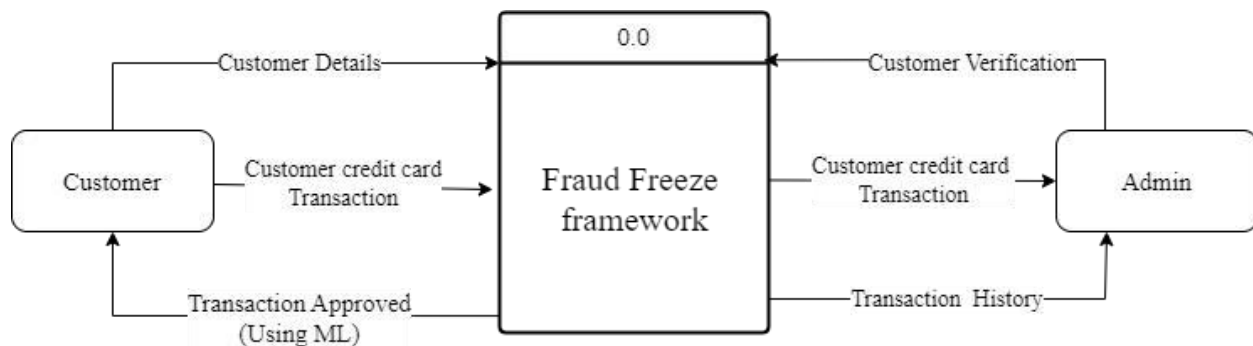
## CHAPTER 5

### SYSTEM DESIGN

#### 5.1 Data Flow Diagram

##### 5.1.1 Data Flow Diagram Level - 0

The Fraud Freeze Framework DFD level 0 depicts the abstract view with the mechanism, represented as a single process with external parties. This DFD for the Fraud Freeze Framework shows the overall structure as a single bubble. It comes with incoming/outgoing indicators showing input and output data as shown in figure 5.1.



**Figure 5.1 Data Flow Diagram Level - 0**

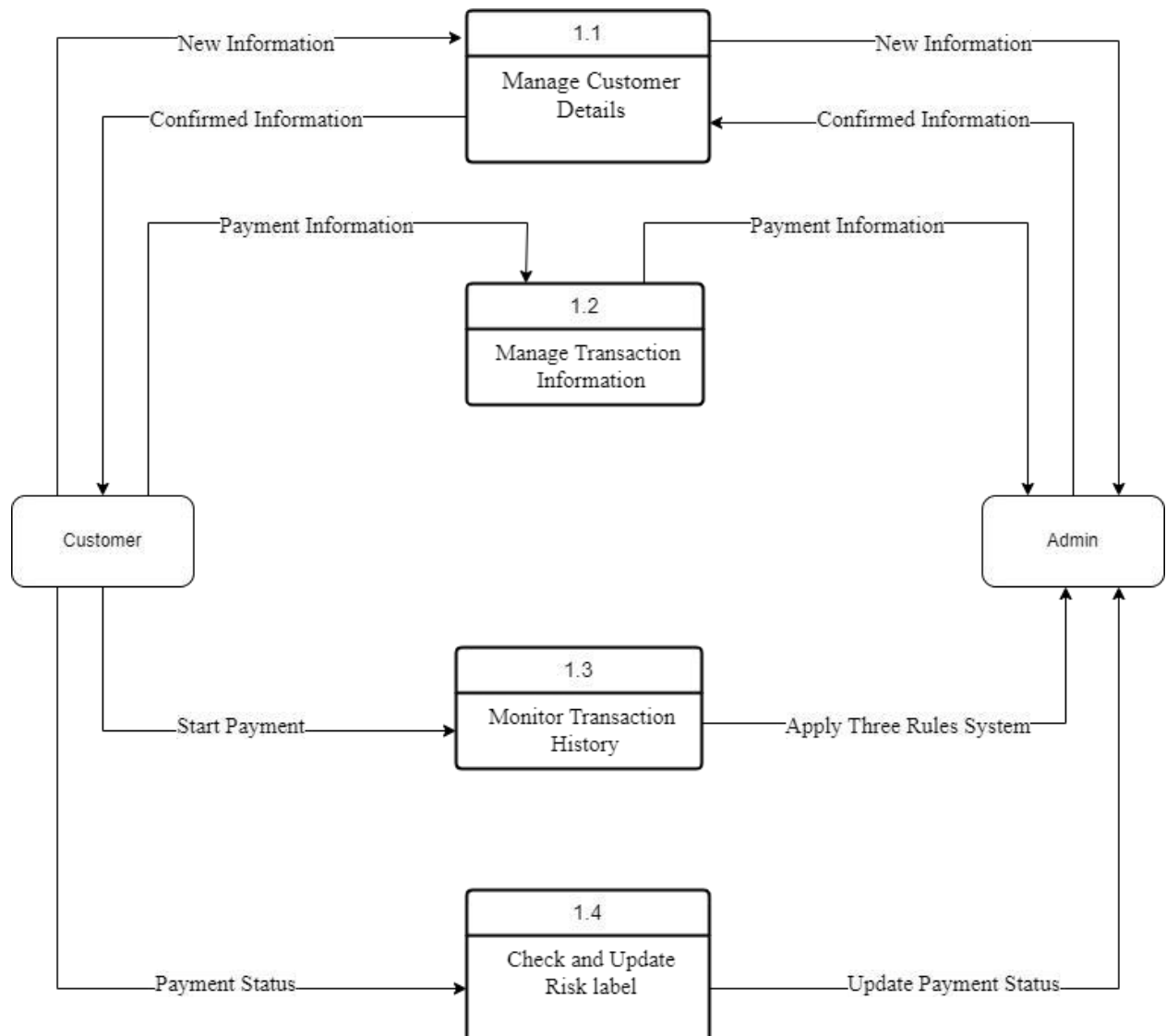
##### 5.1.2 Data Flow Diagram Level - 1

DFD level 1 of Fraud Freeze Framework is a single process node from the level 0 diagram and is broken down into sub processes as shown in figure 5.2. In this level, the system displays the further processing informations.



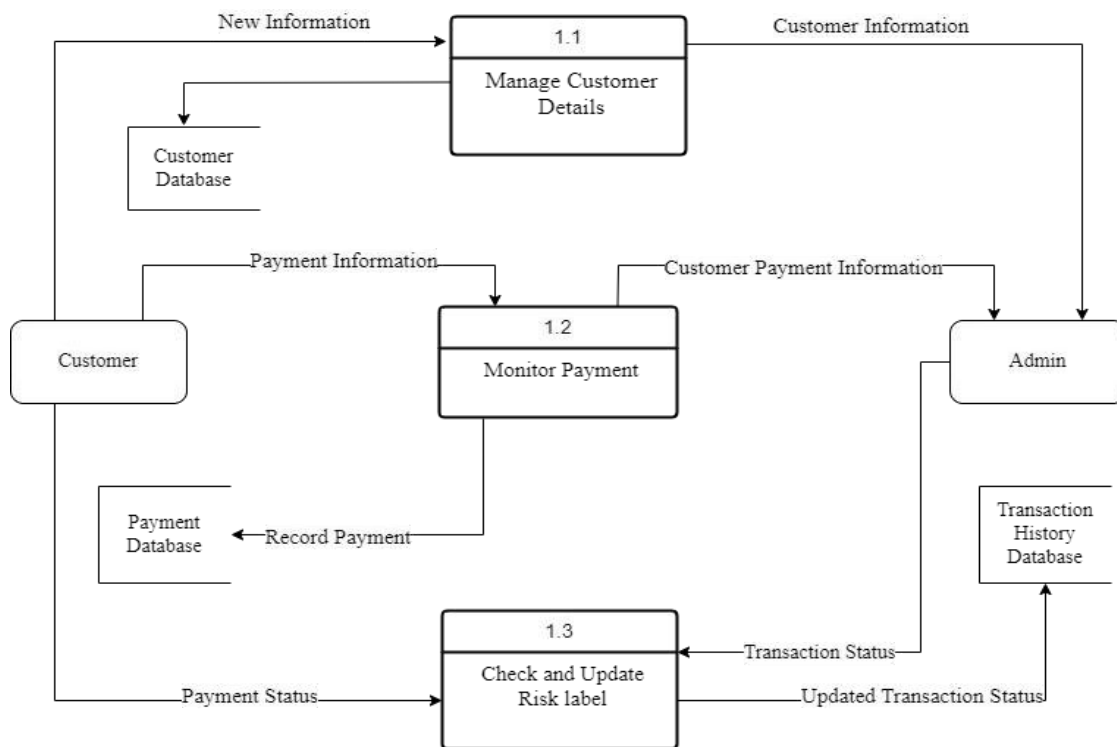
The following are the essential data to accommodate:

- Customer details
- Transaction Information
- Transaction history
- Check and Update Risk label



**Figure 5.2 Data Flow Diagram Level - 1**

### 5.1.3 Data Flow Diagram Level - 2



**Figure 5.3 Data Flow Diagram Level - 2**

The Level 2 DFD for the Fraud Freeze Framework in figure 5.3 represents the basic modules as well as data flow between them. Since the DFD level 2 is the highest abstraction level, Fraud Freeze Framework processes must be detailed and are based on the DFD level 1. Finally, after figuring out the processes given in the system, the user will now have their request being processed. The Processes that the system should prioritize are as follows:

- Manage Customer details
- Manage Transaction Information
- Update Risk label
- Monitor Transaction history
- Manage fraud transactions

## 5.2 CLASS DIAGRAM

Class diagram of Fraud Freeze Framework is the static view of the application as shown in figure 5.4. It describes the attributes and operations of a class and also the constraints imposed on the system. Class diagram shows the collection of classes, interfaces, associations, collaborations, and constraints and it is also known as structural diagram.

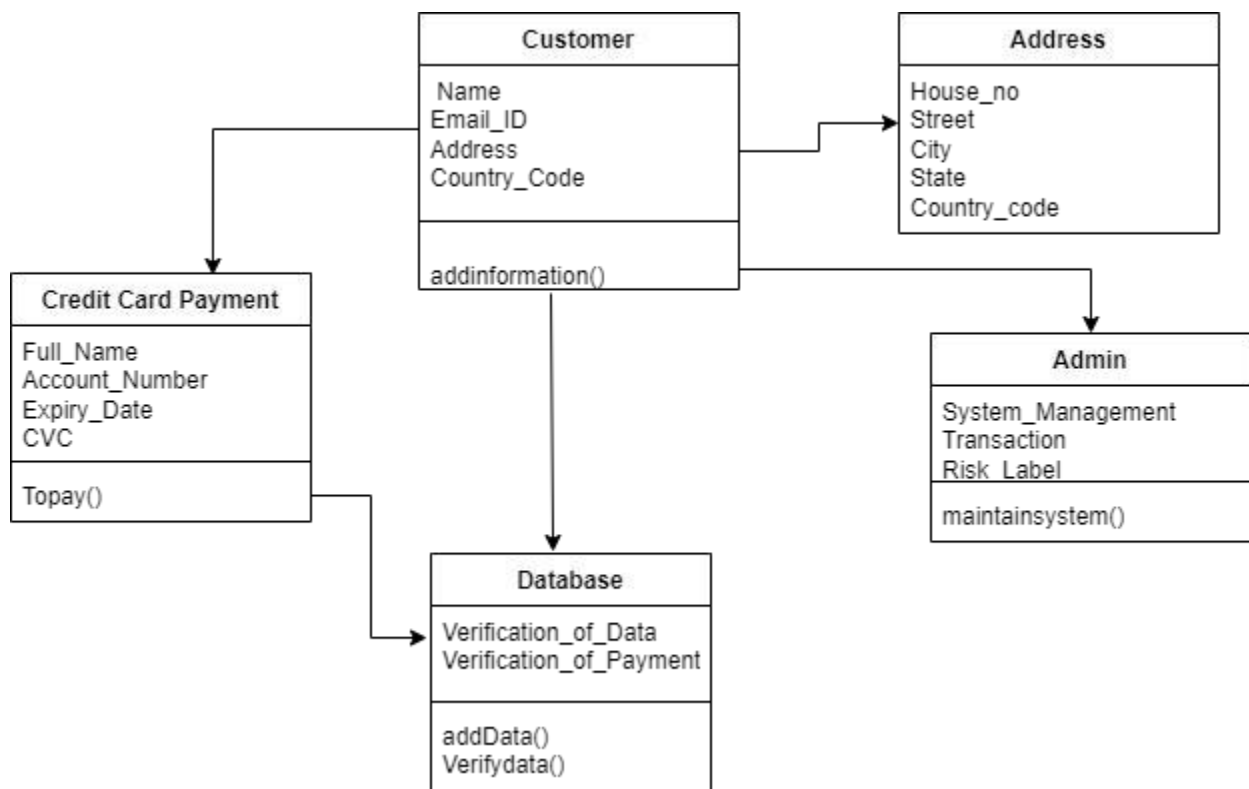


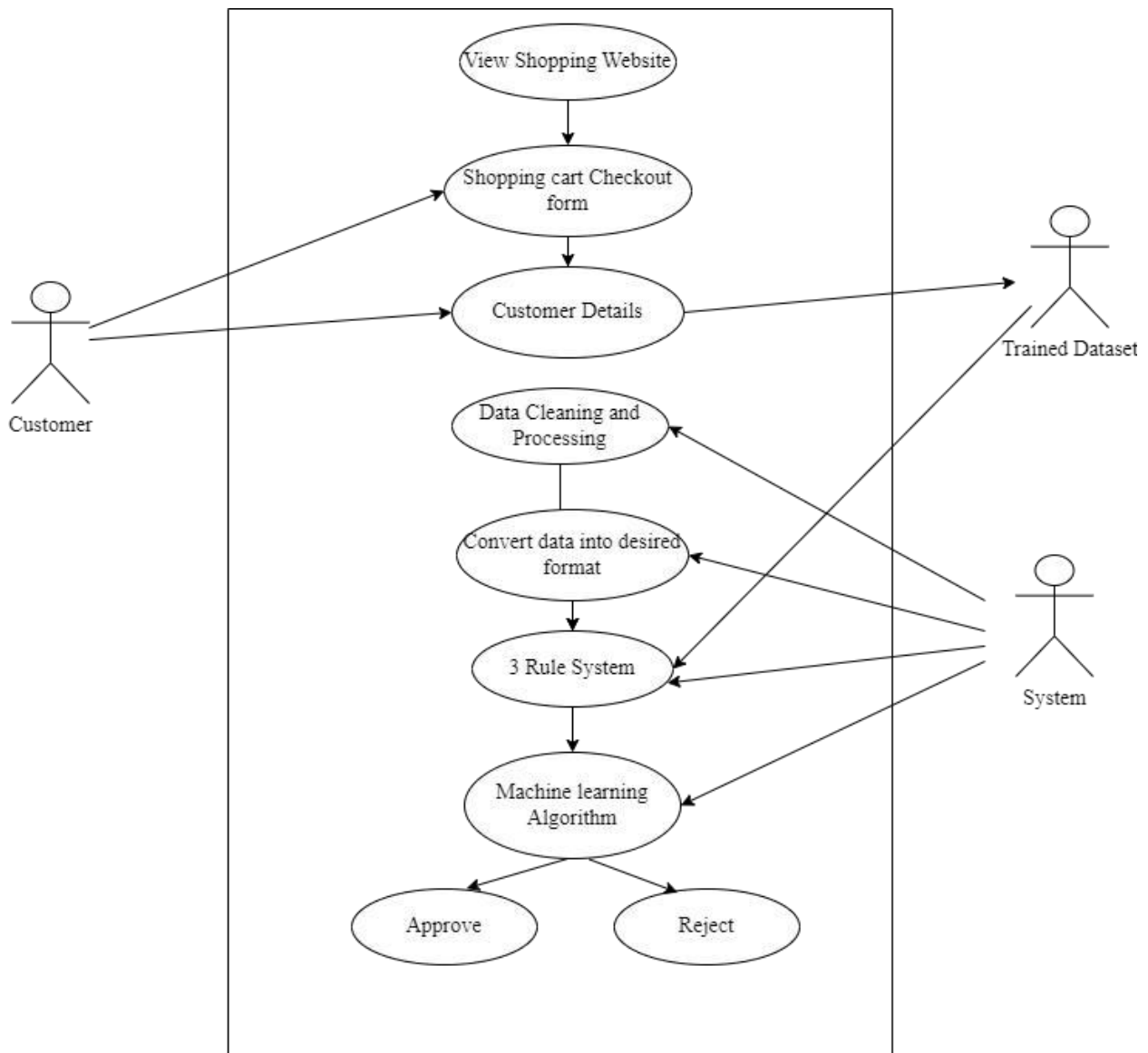
Figure 5.4 Class Diagram

## 5.3 USE CASE DIAGRAM

The use case diagram of Fraud Freeze Framework summarizes the details of the system's users (also known as actors) and their interactions with the system as shown in figure 5.5.

The use case diagram is ideal for:

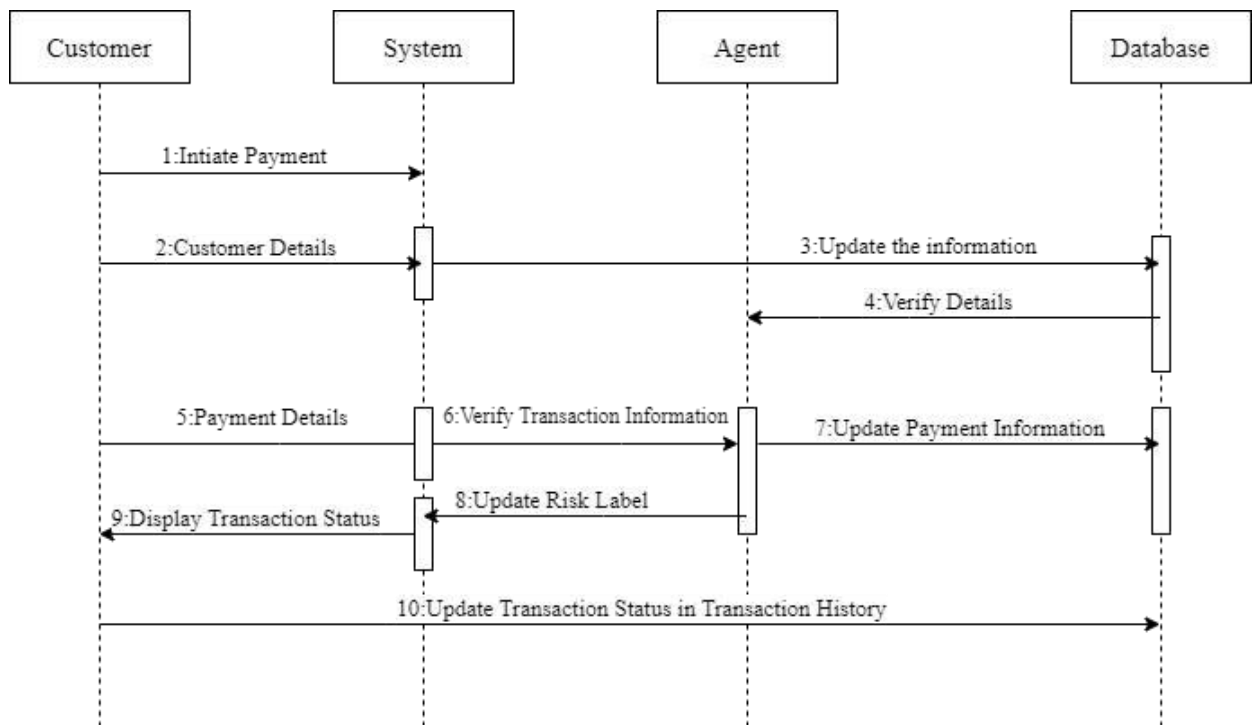
- Representing the goals of system-user interactions
- Defining and organizing functional requirements in a system
- Specifying the context and requirements of a system
- Modeling the basic flow of events in a use case



**Figure 5.5 Use Case Diagram**

## 5.4 SEQUENCE DIAGRAM

The sequence diagram or system sequence diagram (SSD) of Fraud Freeze Framework in figure 5.6, shows the process of interactions arranged in time sequence. It depicts the processes involved and the sequence of messages exchanged between the processes to carry out the functionality. They are sometimes called event diagrams or event scenarios. The parallel vertical lines (lifelines) shows different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.



**Figure 5.6 Sequence Diagram**

## CHAPTER 6

### CONCLUSION

#### 6.1 PERFORMANCE ANALYSIS

The process of model building is not complete without the evaluation of model performance. The most common metric used to evaluate the performance of a classification predictive model is classification accuracy. Typically, the accuracy of a predictive model is good (above 90% accuracy), therefore it is also very common to summarize the performance of a model in terms of the error rate of the model.

- $\text{Accuracy} = \text{Correct Predictions} / \text{Total Predictions}$
- $\text{Error Rate} = \text{Incorrect Predictions} / \text{Total Predictions}$

Classification accuracy is easy to calculate and intuitive to understand, making it the most common metric used for evaluating classifier models. This intuition breaks down when the distribution of examples to classes is severely skewed. Intuitions developed by practitioners on balanced datasets, such as 99 percent representing a skillful model, can be incorrect and dangerously misleading on imbalanced classification predictive modeling problems.

Hence for this project accuracy is not preferred as the performance metric. Log-loss is one of the major metrics to assess the performance of a classification problem. Log-loss is indicative of how close the prediction probability is to the corresponding actual/true value (0 or 1 in case of binary classification).

The more the predicted probability diverges from the actual value, the higher is the

log-loss value. The machine learning models used are,

- Logistic Regression
- XGBoost Classifier
- K-Nearest Neighbour
- Gradient Boosting Classifier

These models are analyzed using Ensemble Methods like Averaging, Max voting, Stacking and Bagging. In Averaging, Max voting, Stacking, three classification models (Logistic Regression, XGBoost and K-Nearest Neighbour) are combined and the model is trained. In Bagging, K-Nearest Neighbour algorithm is used as the base model. Confusion matrix, accuracy, precision, recall, F1-score and support of both training and testing dataset are recorded for each model below.

### Averaging Method

TRAINING RESULTS:					
=====					
CONFUSION MATRIX:					
[[80569 3153]					
[ 794 82928]]					
ACCURACY SCORE:					
0.9764					
CLASSIFICATION REPORT:					
	0	1	accuracy	macro avg	weighted avg
precision	0.990241	0.963372	0.976428	0.976806	0.976806
recall	0.962340	0.990516	0.976428	0.976428	0.976428
f1-score	0.976091	0.976755	0.976428	0.976423	0.976423
support	83722.000000	83722.000000	0.976428	167444.000000	167444.000000
TESTING RESULTS:					
=====					
CONFUSION MATRIX:					
[[19134 1948]					
[ 85 2174]]					
ACCURACY SCORE:					
0.9129					
CLASSIFICATION REPORT:					
	0	1	accuracy	macro avg	weighted avg
precision	0.995577	0.527414	0.9129	0.761496	0.950267
recall	0.907599	0.962373	0.9129	0.934986	0.912900
f1-score	0.949555	0.681398	0.9129	0.815476	0.923602
support	21082.000000	2259.000000	0.9129	23341.000000	23341.000000

## Max Voting

```
TRAINIG RESULTS:
=====
CONFUSION MATRIX:
[[80569 3153]
 [ 794 82928]]
ACCURACY SCORE:
0.9764
CLASSIFICATION REPORT:
              0              1  accuracy  macro avg  weighted avg
precision    0.990241    0.963372  0.976428    0.976806    0.976806
recall       0.962340    0.990516  0.976428    0.976428    0.976428
f1-score     0.976091    0.976755  0.976428    0.976423    0.976423
support      83722.000000  83722.000000  0.976428  167444.000000  167444.000000
TESTING RESULTS:
=====
CONFUSION MATRIX:
[[19134 1948]
 [  85 2174]]
ACCURACY SCORE:
0.9129
CLASSIFICATION REPORT:
              0              1  accuracy  macro avg  weighted avg
precision    0.995577    0.527414  0.9129    0.761496    0.950267
recall       0.907599    0.962373  0.9129    0.934986    0.912900
f1-score     0.949555    0.681398  0.9129    0.815476    0.923602
support      21082.000000  2259.000000  0.9129  23341.000000  23341.000000
```

## Stacking Method

```
TRAINIG RESULTS:
=====
CONFUSION MATRIX:
[[83722  0]
 [  0 83722]]
ACCURACY SCORE:
1.0000
CLASSIFICATION REPORT:
              0              1  accuracy  macro avg  weighted avg
precision    1.0      1.0      1.0      1.0      1.0
recall       1.0      1.0      1.0      1.0      1.0
f1-score     1.0      1.0      1.0      1.0      1.0
support      83722.0  83722.0      1.0  167444.0    167444.0
TESTING RESULTS:
=====
CONFUSION MATRIX:
[[21082  0]
 [  0 2259]]
ACCURACY SCORE:
1.0000
CLASSIFICATION REPORT:
              0              1  accuracy  macro avg  weighted avg
precision    1.0      1.0      1.0      1.0      1.0
recall       1.0      1.0      1.0      1.0      1.0
f1-score     1.0      1.0      1.0      1.0      1.0
support      21082.0  2259.0      1.0  23341.0    23341.0
```



## Bagging Method

```
TRAINING RESULTS:
=====
CONFUSION MATRIX:
[[75122  8600]
 [ 8176 75546]]
ACCURACY SCORE:
0.8998
CLASSIFICATION REPORT:
              0              1  accuracy  macro avg  weighted avg
precision    0.901846    0.897797  0.899811    0.899822    0.899822
recall       0.897279    0.902343  0.899811    0.899811    0.899811
f1-score     0.899557    0.900064  0.899811    0.899811    0.899811
support      83722.000000  83722.000000  0.899811  167444.000000  167444.000000
TESTING RESULTS:
=====
CONFUSION MATRIX:
[[16548  4534]
 [ 1017  1242]]
ACCURACY SCORE:
0.7622
CLASSIFICATION REPORT:
              0              1  accuracy  macro avg  weighted avg
precision    0.942101    0.215028  0.762178    0.578564    0.871733
recall       0.784935    0.549801  0.762178    0.667368    0.762178
f1-score     0.856367    0.309147  0.762178    0.582757    0.803405
support      21082.000000  2259.000000  0.762178  23341.000000  23341.000000
```

## Comparison of training and test dataset accuracy

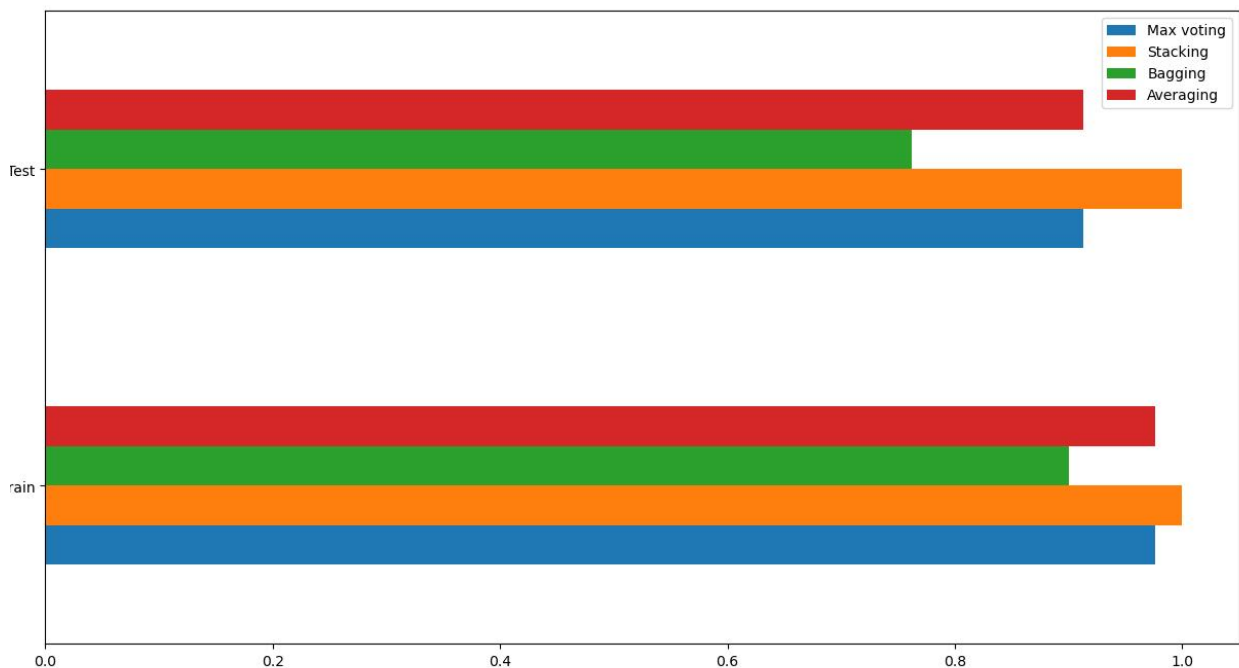


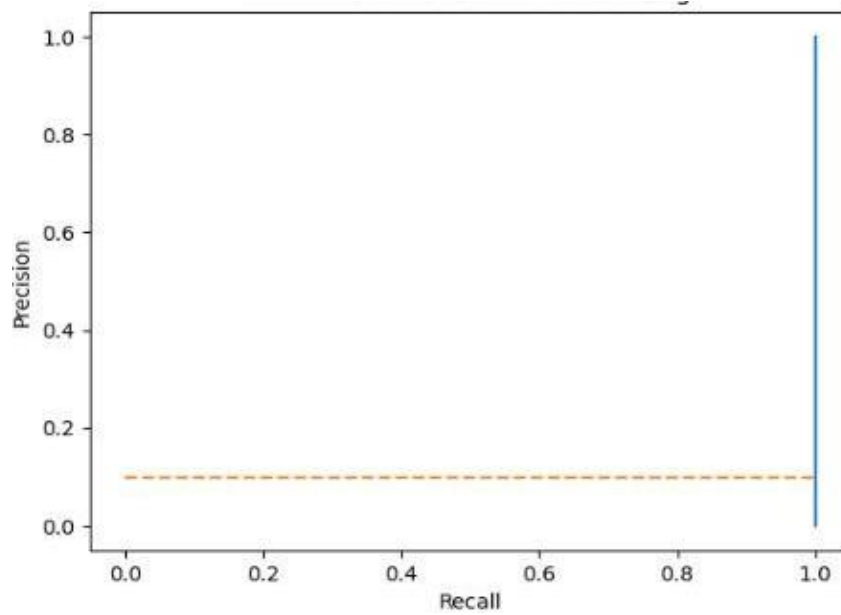
Figure 6.1 Comparison of accuracy of each ensemble model

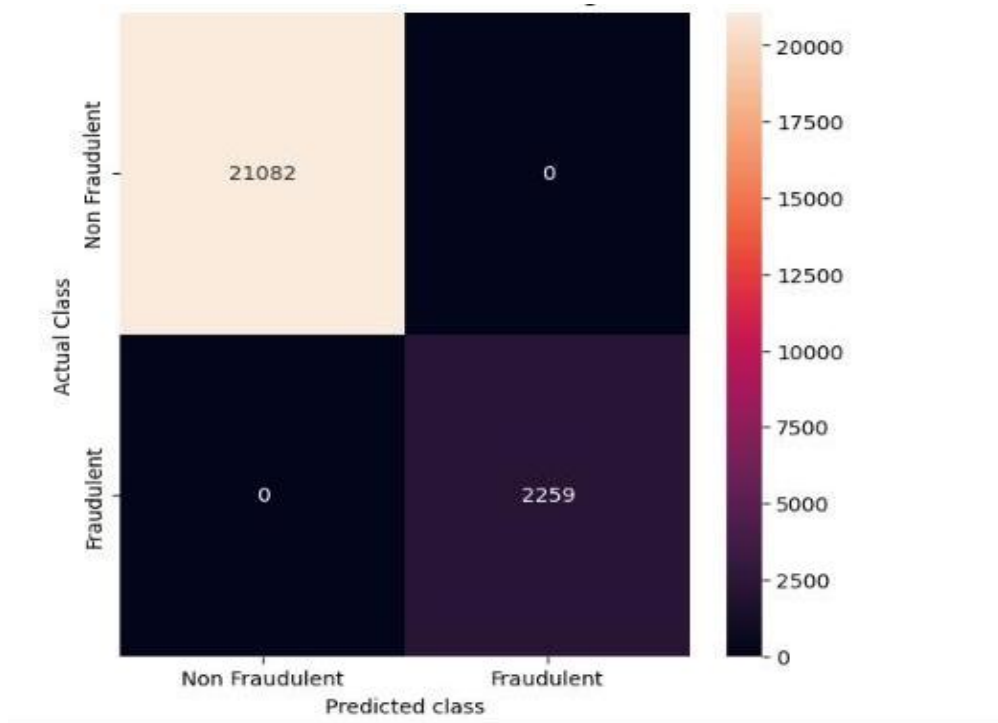
Thus log-loss was calculated for each ensemble model and the results obtained are recorded in table 6.1.

**Table 6.1 Log-Loss of Ensemble Models**

	<b>esemble_method</b>	<b>log-loss</b>
2	Stacking	9.992007e-16
0	Averaging	2.824487e-01
1	Max Voting	3.008393e+00
3	Bagging	8.214231e+00

Thus, Stacking method has the least log-loss and better accuracy compared to other methods as seen in figure 6.1. Thus it is the highest performing model. The Precision Recall Curve and the Confusion matrix of Stacking method is shown in figure 6.2.





**Figure 6.2 Precision Recall Curve and Confusion Matrix of Max Voting Model**

## 6.2 RESULTS AND DISCUSSIONS

Credit card fraud is an increasing threat to financial institutions. Fraudsters tend to constantly come up with new fraud methods. A robust classifier can only handle the changing nature of fraud. Accurately predicting fraud cases and reducing false-positive cases is the foremost priority of a fraud detection system.

The performance of ML methods varies for each individual business case. The type of input data is a dominant factor that drives different ML methods. For detecting credit card fraud, the number of features, number of transactions, and correlation between the features are essential factors in determining the model's performance.

Ensemble methods, such as stacking method can harness the capabilities of a range of well-performing models on a classification task and make predictions that have better performance than any single model in the ensemble. Using this model for the detection of credit card fraud detection yields better performance than traditional algorithms. Comparing all the ensemble method performances side to side, Stacking with Logistic Regression, XGBoost and K-Nearest Neighbour as the baseline models, is the top method with minimum log loss of  $9.992007e-16$  and maximum accuracy of 100%.

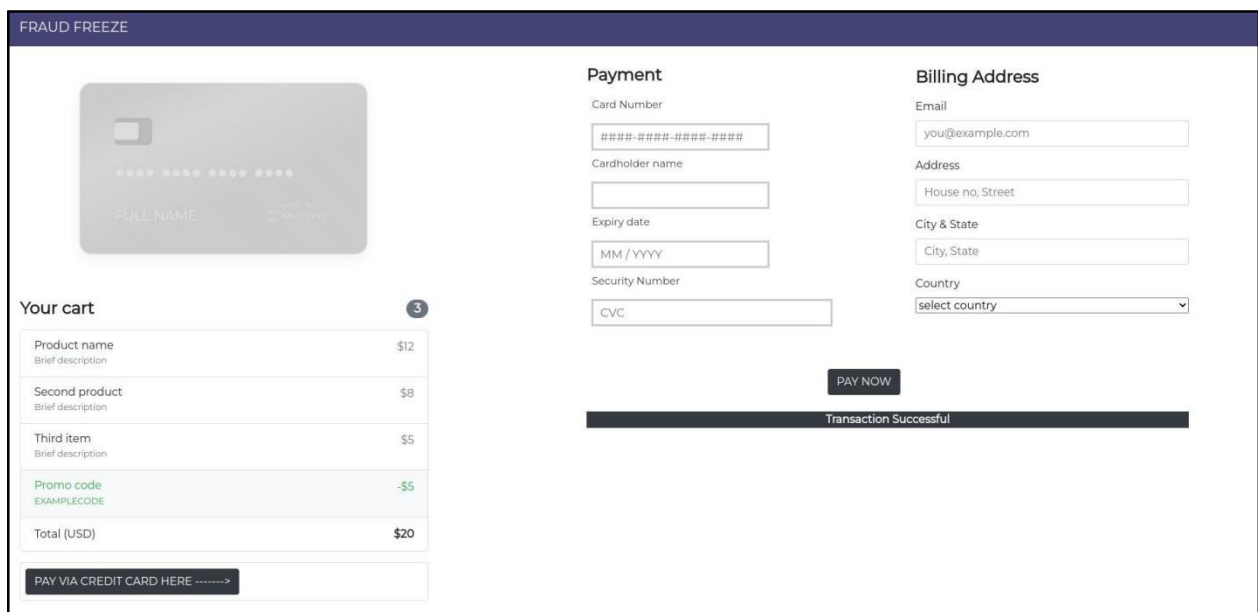
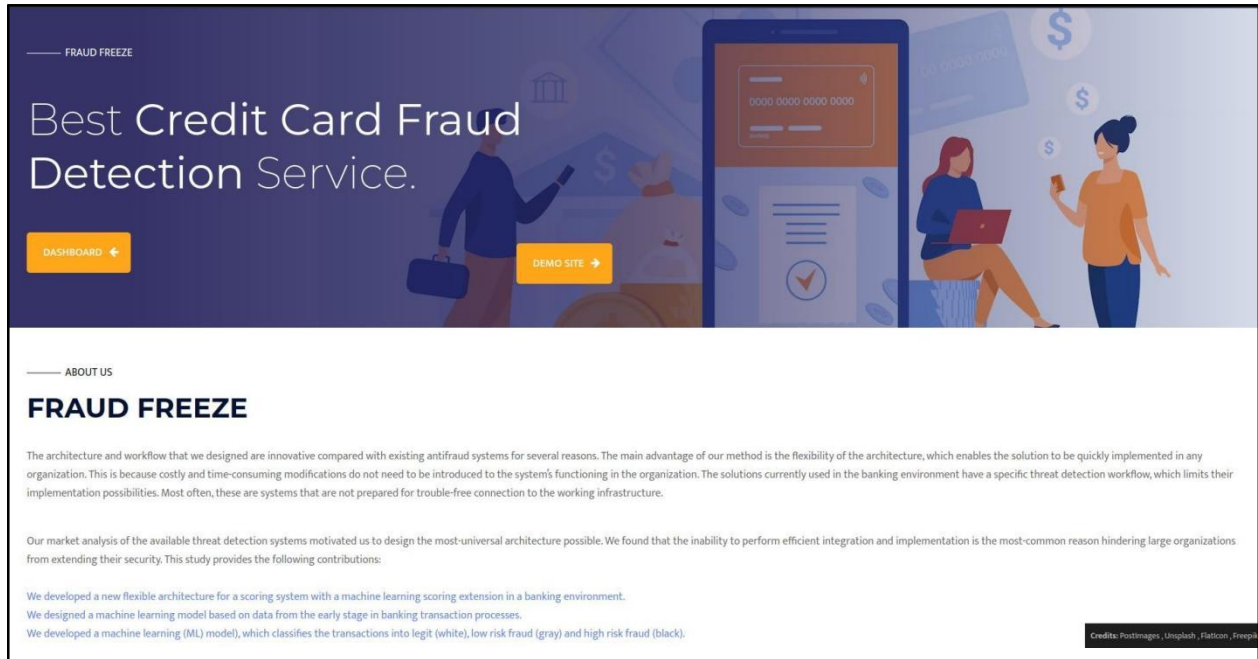
### **6.3 FUTURE ENHANCEMENTS**

This project has explained in detail how machine learning can be applied to get better results in fraud detection along with the pre-processing of the data, implementation and experimentation results. This project can be further improved by integrating multiple algorithms together as modules and their results can be also combined to increase the accuracy of the final result. This would provide a great degree of modularity and versatility to the framework. The precision of the algorithms increases when the size of the dataset is increased.

Hence, more data will surely make the model more accurate in detecting frauds and reduce the number of false positives. However, this requires official support from the banks themselves, to obtain large real-time dataset. With the increasing maturity of privacy preserving technologies, i.e. the ability to share insights without sharing the data, involved parties such as banks, payment service providers, etc. will become more open to contributing to the collective intelligence network. This will enhance training of ML models and optimization algorithms with the necessary data and therefore prevent fraudulent behavior more efficiently.

# APPENDICES

## A. SCREENSHOTS OF THE DEMONSTRATION



FRAUD FREEZE

### Your cart

Product name Brief description	\$12
Second product Brief description	\$8
Third item Brief description	\$5
Promo code EXAMPLECODE	-\$5
Total (USD)	\$20

PAY VIA CREDIT CARD HERE ----->

### Payment

Card Number  
#####-####-####

Cardholder name  
\_\_\_\_\_

Expiry date  
MM / YYYY

Security Number  
CVC

### Billing Address

Email  
you@example.com

Address  
House no, Street

City & State  
City, State

Country  
select country

PAY NOW

Transaction Failed - Contact Admin

Dashboard

Analytics

Fraud Freeze Dashboard

Logout

Amount Transacted  
**\$750.90**  
↑ 13% Since last month

New Users  
**215**  
↑ 34% Since last month

Total hours  
**1.400**  
↓ -9% Since last month

Work load  
**95%**  
↑ 10% Since last month

NAME	IP ADDRESS	TRANSACTION TIME	EMAIL-ID	RISK LABEL	
Ana	103.182.69.43	11-03-2023	anajessica@gmail.com	Legit	Transaction Successful
Ana	103.182.69.43	11-03-2023	anajessica@gmail.com	Legit	Transaction Successful
Ana	103.182.69.43	11-03-2023	anajessica@gmail.com	Legit	Transaction Successful
Ana	100.182.69.43	11-03-2023	anajessica@gmail.com	High risk	Alert User
Febi	103.182.69.22	2023-03-12 02:38:00.263432	anajessica15022002@gmail.com	Low risk	Alert User
Febi	103.182.69.45	2023-03-12 18:52:46.728018	anajessica15022002@gmail.com	Low risk	Alert User
Febi	103.182.69.45	2023-03-12 18:53:49.196611	anajessica15022002@gmail.com	Low risk	Alert User
Anitha	103.182.69.45	2023-03-12 18:59:49.962425	anajessica15022002@gmail.com	High risk	Alert User

## REFERENCES

- [1]Bora Mehar Sri Satya Teja, Boomireddy Munendra and Mr. S. Gokulkrishnan, “A Research Paper on Credit Card Fraud Detection”, in International Research Journal of Engineering and Technology (IRJET), Volume: 09, Issue: 03, Mar 2022 p-ISSN: 2395-0072.
- [2]Emmanuel Illeberi, Yanxia Sun and Zenghui Wang, in “Performance Evaluation of Machine Learning Methods for Credit Card Fraud Detection Using SMOTE and AdaBoost” in December 2021 IEEE, Volume 9, 2021, pp. 165-294.
- [3]Lakshmi S V S S and Selvani Deepthi Kavila, “Machine Learning For Credit Card Fraud Detection System”. in International Journal of Applied Engineering Research. ISSN 0973-4562, Volume 13, Number 24 (2018) pp. 16819-16824.
- [4]Mosa M. M. Megdad, Bassem S. Abu-Nasser and Samy S. Abu-Naser, “Fraudulent Financial Transactions Detection Using Machine Learning”, in International Journal of Academic Information Systems Research (IJASIR), ISSN: 2643-9026, Vol. 6, Issue 3, March - 2022.
- [5]Shayan Wangde, Raj Kheratkar, Zoheb Waghu and Prof. Suhas Lawand, “Online Transaction Fraud Detection System Using Machine Learning & E-Commerce”, in International Research Journal of Engineering and Technology (IRJET), e-ISSN: 2395-0056, Volume: 09, Issue: 04, Apr 2022.
- [6]Fawaz Khaled Alarfaj, Iqra Malik, Hikmat Ullah Khan, Naif Almusallam, Muhammad Ramzan, and Muzamil Ahmed, “Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms”, in IEEE, Volume 10, 2022, pp. 39715.