EFFICIENT PAYMENT FRAUD DETECTION WITH META LEARNING CLASSIFIER IN A COST-SENSITIVE FRAMEWORK

A "Project Stage II" Report submitted to JNTU Hyderabad in partial fulfilment of the requirements for the award of the degree

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

RAPELLY SAHITH	**21S11A05G9**
SINGIREDDY MANOJ REDDY	**21S11A05F5**
EERLA ADARSH	**21S11A05D2**
MUTHYALA SAI PRANAY	**21S11A05H3**

Under the Guidance of

Mrs. ANURADHA REDDY
Associate Professor of CSE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MALLA REDDY INSTITUTE OF TECHNOLOGY & SCIENCE
(Approved by AICTE New Delhi and Affiliated to JNTUH)
(Accredited by NBA & NAAC with "A" Grade)
An ISO 9001: 2015 Certified Institution
Maisammaguda, Medchal (M), Hyderabad-500100, T. S.

APRIL 2025

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MALLA REDDY INSTITUTE OF TECHNOLOGY & SCIENCE

(Approved by AICTE New Delhi and Affiliated to JNTUH)

(Accredited by NBA & NAAC with "A" Grade)

An ISO 9001: 2015 Certified Institution

Maisammaguda, Medchal (M), Hyderabad-500100, T. S.

APRIL 2025

CERTIFICATE

This is to certify that the "Project Stage II" entitled **"EFFICIENT PAYMENT FRAUD DETECTION WITH META LEARNING CLASSIFIER IN A COST-SENSITIVE FRAMEWORK"** has been **submitted by RAPELLY SAHITH (21S11A05G9), MANOJ REDDDY (21S11A05F5), EERLA ADARSH (21S11A05D2) and MUTHYALA SAI PRANAY (21S11A05H3)** in partial fulfillment of the requirements for the award of **BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING.** This record of Bonafide work carried out by them under my guidance and supervision. **The result embodied in this Project Stage II report has not been submitted to any other University or Institute for the award of any degree.**

Mrs. ANURADHA REDDY
Associate Professor of CSE
Project Guide

Mrs. K. MAMATHA
Head of the Department

External Examiner

ACKNOWLEDGEMENT

The Project Stage II work carried out by our team in the Department of Computer Science and Engineering, Malla Reddy Institute of Technology and Science, Hyderabad. ***This work is original and has not been submitted in part or full for any degree or diploma of any other university.***

We wish to acknowledge our sincere thanks to our project guide **Mrs. Anuradha Reddy,** Associate Professor of Computer Science & Engineering for formulation of the problem, analysis, guidance and his continuous supervision during the course of work.

We acknowledge our sincere thanks to **Dr. Vaka Murali Mohan,** Principal and **Mrs. K. Mamatha**, Head of the Department and Coordinator, faculty members of CSE Department for their kind cooperation in making this Project Stage II work a success.

We extend our gratitude to **Sri.Ch. Malla Reddy**, Founder Chairman MRGI and **Sri. Ch. Mahender Reddy,** Secretary MRGI, **Dr. Ch. Bhadra Reddy**, President MRGI, **Sri.Ch. Shalini Reddy**, Director MRGI, **Sri. P. Praveen Reddy**, Director MRGI, for their kind cooperation in providing the infrastructure for completion of our Project Stage II

We acknowledge our special thanks to the entire teaching faculty and non-teaching staff members of the Computer Science & Engineering Department for their support in making this project work a success.

RAPELLY SAHITH	21S11A05G9	_____
SINGIREDDY MANOJ REDDY	21S11A05F5	_____
EERLA ADARSH	21S11A05D2	_____
MUTHYALA SAI PRANAY	21S11A05H3	_____

INDEX

ABSTRACT

The act of fraudulent credit card transaction has been increased over the past recent years, as the era of digitization hits our day-to-day life, with people are getting more involved in online banking and online transaction system. Machine learning algorithms have played a significant role in detection of credit card frauds. However, the unbalanced nature of the real-life datasets causes the traditional classification algorithms to perform low in detection of credit card fraud. In this work, a cost-sensitive weighted random forest algorithm has been proposed for effective credit card fraud detection. A cost-function has been defined in the training phase of each tree, in bagging which emphasizes to assign more weight to the minority instances during training. The trees are ranked according to their predictive ability of the minority class instances. The proposed work has been compared with two existing random-forest based techniques for two binary credit card datasets. The efficiency of the model has been evaluated in terms G-mean, F-measure and AUC values. The experimental results have established the proficiency of the proposed model, than the existing ones.

LIST OF FIGURES

CHAPTER-1: SYSTEM ANALYSIS

1.1 EXISTING SYSTEM

However, the chances of achieving adequate result is still questionable as the cases of imbalanced data are present, in each tree. So, to improve the prediction ability of each tree, and the overall performance of the ensemble, a cost- function is defined, based on the misclassification rate of the instances, both majority and minority. The trees are given weightage based on the quantity of error; they are yielding to the ensemble. The trees with lower error-rate are given higher weightage while determining the final outcome of the test object.

The chances of achieving adequate result is still questionable as the cases of imbalanced data are present, in each tree. So, to improve the prediction ability of each tree, and the overall performance of the ensemble, a cost- function is defined, based on the misclassification rate of the instances, both majority and minority. The trees are given weightage based on the quantity of error; they are yielding to the ensemble. The trees with lower error-rate are given higher weightage while determining the final outcome of the test object.

1.1.1 DISADANTAGES

The existing systems for payment fraud detection face several significant disadvantages. One major challenge is dealing with highly imbalanced datasets, where fraudulent transactions represent only a small fraction of the total, making it difficult for traditional methods to accurately detect fraud without compromising the performance on legitimate transactions. Additionally, these systems often have a high false positive rate, flagging legitimate transactions as fraudulent, which not only frustrates customers but also increases the cost of manual investigations. Many existing systems are not adaptive enough to handle the rapidly evolving nature of fraud, as fraudsters continuously develop new techniques to bypass detection. Furthermore, traditional methods often lack cost-sensitive frameworks, failing to appropriately balance the financial impact of missed frauds (false negatives) and unnecessary alerts (false positives). Overfitting to historical data is another issue, as it prevents these models from generalizing effectively to new patterns of fraud. Moreover, some existing solutions are computationally expensive, making real-time fraud detection challenging, and they may struggle with scalability as transaction volumes increase. These limitations highlight the need for more advanced approaches, such as the integration of meta-learning classifiers within cost-sensitive frameworks, to address these shortcomings effectively.

1.2 PROPOSED SYSTEM

In this paper, the imbalanced nature of the credit card data has been taken into consideration, and a cost-sensitive weighted random forest technique has been proposed to overcome the issue. The Random Forest (RF) algorithm is basically an ensemble learning technique which works on the principle of Bagging, i.e. the dataset has been divided into n-bags or samples, with randomly selected instances, and each of which is termed as "Tree" The final output of the model is achieved based on majority voting (in case of classification model), or averaging (in case of regression model) of all the test outcomes, with respect to all the trained trees. The foremost advantage of using RF technique for credit card fraud detection is that it can readily deal with the enormous size the training data, as RF involves to divide the dataset into a number samples, and then learn. Through the proposed approach, a cost- driven learning scheme is adapted to give more emphasis on learning the minority class instances. The rest of the paper is organized as follows. In the proposed scheme, a cost- sensitive weighted RF algorithm has been proposed for credit card fraud detection. A confusion -matrix based cost-function is defined to put more emphasis on the minority class instances during training. Instead of conventional majority voting strategy, a weighing strategy, based on minimum error achieved for a single tree.

1.2.1 ADVANTAGES

The proposed system in offers several advantages over existing methods. By incorporating a meta-learning classifier, the system adapts dynamically to changing fraud patterns, ensuring improved accuracy and robustness in detecting fraudulent transactions. The cost-sensitive framework prioritizes reducing financial losses by balancing the trade-off between false positives and false negatives, thereby addressing the limitations of traditional cost-insensitive models. This approach enhances detection capabilities for the minority class (fraudulent transactions) in highly imbalanced datasets, minimizing the risk of missed frauds without significantly increasing false alarms. Additionally, the proposed system is computationally efficient, making it suitable for real-time fraud detection in high-volume transaction environments. Its scalability ensures consistent performance as the transaction load increases. Overall, the combination of meta-learning and cost-sensitive principles delivers a more reliable, adaptive, and economically viable solution for payment fraud detection.

1.3 INTRODUCTION

Payment card fraud leads to heavy annual financial losses around the world, thus giving rise to the need for improvements to the fraud detection systems used by banks and financial institutions. In the academe, as well, payment card fraud detection has become an important research topic in recent years. With these considerations in mind, we developed a method that involves two stages of detecting fraudulent payment card transactions. The extraction of suitable transactional features is one of the key issues in constructing an effective fraud detection model. In this method, additional transaction features are derived from primary transactional data. A better understanding of cardholders spending behaviors is created by these features. Accordingly, a new similarity measure is established on the basis of transaction time in this stage. This measure assigns greater weight to recent transactions. In the second stage, the dynamic random forest algorithm is employed for the first time in initial detection, and the minimum risk model is applied in cost-sensitive detection. We tested the proposed method on a real transactional dataset obtained from a private bank. The results showed that the recent behavior of cardholders exerts a considerable effect on decision-making regarding the evaluation of transactions as fraudulent or legitimate. The findings also indicated that using both primary and derived transactional features increases the F-measure. Finally, an average 23% increase in prevention of damage (Pod) is achieved with the proposed cost-sensitive approach. The term credit card fraud signifies any act of theft and fraud, occurred in case of any payment card (debit or credit), due to physical loss of the card by the owner, or stealing of the card by fraudsters, or by means techniques like phishing, skimmer, identity theft etc. As discussed in there are two types of credit card fraud can happen, internal fraud and external fraud. The first type depicts a situation where there is leakage of information between the cardholder and bank, by means of false identity; whereas in the later case, fraud happens due to stolen or lost credit card get into some fraudsters' hands. To solve this problem, credit card fraud detection techniques have been developed by researchers over the past years. Basically, it involves of classifying the credit card transactions either as "legitimate" or "fraudulent". A number of Machine Learning (ML) techniques have been employed for this task, such as Decision tree Support Vector Machine (SVM), Naïve Bayes (NB), Artificial Neural network (ANN), and optimization techniques such as genetic algorithm, migrating bird's optimization algorithm. However, the task of credit card fraud has to face some adversities such as, (a) unavailability of real-life credit transaction dataset, (b) imbalanced ratio of "legitimate" vs. "fraudulent" transactions, (c) enormous size of the dataset, and (d) dynamic behavior of the fraudsters. As a result, effective credit card fraud detection demands effective pre-processing of the dataset, prior to applying of any ML techniques... Each tree is then trained by using a weak-classifier (C4.5/J48), and then the test set is validated with trained trees.

CHAPTER 2: LITERATURE SURVEY

[1] "Credit card fraud detection using Machine Learning Techniques"

J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare.

Credit card fraud detection is a critical issue in the financial sector, as it directly impacts both consumers and financial institutions. The increasing volume of online transactions has made it more challenging to detect fraudulent activities promptly. This paper explores the application of machine learning (ML) techniques to enhance the accuracy and efficiency of credit card fraud detection systems. Various ML algorithms, such as decision trees, support vector machines (SVM), random forests, and neural networks, are analyzed and compared based on their performance in detecting fraudulent transactions. The study also discusses the challenges of imbalanced datasets, feature engineering, and the use of anomaly detection methods to improve detection rates. By leveraging these techniques, the proposed model aims to minimize false positives while maximizing fraud detection accuracy. The results demonstrate that machine learning can significantly improve the effectiveness of fraud detection systems, offering a promising solution to reduce financial losses and enhance the security of credit card transactions.

[2] "Credit Card Fraud Detection Using Decision Tree Induction Algorithm"

S. Patil, H. Somavanshi, J. Gaikwad, A. Deshmane.

Credit card fraud detection is an essential aspect of ensuring security in financial transactions, as fraudulent activities can lead to significant financial losses for both consumers and financial institutions. This paper proposes the use of the Decision Tree Induction Algorithm for detecting fraudulent credit card transactions. Decision trees are a popular machine learning technique known for their simplicity, interpretability, and effectiveness in classification tasks. The paper explores how the Decision Tree Induction Algorithm can be applied to identify patterns and anomalies in transaction data, distinguishing between legitimate and fraudulent transactions. The study also addresses challenges such as the imbalance in fraud detection datasets and how to mitigate issues like overfitting and underfitting. Experiments with real-world credit card transaction data demonstrate the effectiveness of decision trees in providing high detection accuracy, low false positives, and efficient computation. The results suggest that the Decision Tree Induction Algorithm is a promising tool for improving fraud detection systems, offering both accuracy and interpretability in real-time applications.

[3] "A Machine Learning Approach for Detection of Fraud based on SVM"

G. Singh, R. Gupta, A. Rastogi, M. D. S. Chandel, and A. Riyaz.

Fraud detection, particularly in financial transactions, is a growing concern for institutions and consumers alike. With the increasing complexity of fraudulent activities, traditional methods are becoming less effective. This paper presents a machine learning approach using Support Vector Machine (SVM) for the detection of fraudulent transactions. SVM is a robust and effective classifier that has demonstrated excellent performance in high-dimensional spaces and imbalanced datasets, making it suitable for fraud detection tasks. The study investigates the application of SVM for classifying credit card transactions as either legitimate or fraudulent based on features such as transaction amount, time, location, and user behavior. Key challenges, including handling class imbalance, are addressed by employing techniques like kernel functions and class-weight adjustments. Experimental results on real-world transaction data show that the SVM model outperforms traditional methods, achieving high accuracy, precision, and recall rates in identifying fraud while minimizing false positives. The paper highlights the potential of SVM in providing an efficient and scalable solution for real-time fraud detection in financial systems.

[4]"Improving credit card fraud detection with calibrated probabilities. In Proceedings of the 2020 SIAM International Conference on Data Mining"

A. C. Bahnsen, A. Stojanovic, D. Aouada, and B. Ottersten.

Credit card fraud detection remains a critical challenge for financial institutions, given the increasing sophistication of fraudulent activities. Traditional methods often face limitations in accurately identifying fraudulent transactions while minimizing false positives. This paper proposes an approach to enhance fraud detection by improving the calibration of predicted probabilities in machine learning models. The study focuses on the use of calibrated probabilities to adjust the outputs of classification models, such as decision trees and support vector machines (SVM), to provide more reliable estimates of fraud likelihood. By applying calibration techniques, including Platt scaling and isotonic regression, we demonstrate how to enhance the accuracy and effectiveness of fraud detection systems. Experimental results on real-world credit card transaction data reveal that calibration significantly improves the precision, recall, and overall performance of fraud detection models. The calibrated models achieve better alignment between predicted and actual fraud rates, leading to a reduction in false positives and false negatives. This work underscores the importance of calibrated probability outputs in achieving a more robust and reliable credit card fraud detection system, contributing to enhanced security and operational efficiency in financial transactions.

[5] "Fraud Detection of Credit Card Payment System by Genetic Algorithm."

K. RamaKalyani, and D. UmaDevi.

The detection of fraud in credit card payment systems is a significant challenge due to the increasing complexity and volume of transactions. This paper proposes the use of a Genetic Algorithm (GA) to enhance the accuracy and efficiency of fraud detection in credit card payment systems. Genetic Algorithms, inspired by the principles of natural selection, are employed to optimize the feature selection and classification process in identifying fraudulent transactions. The study explores how GA can be utilized to evolve optimal solutions for selecting relevant features and tuning model parameters, ensuring the detection of subtle patterns indicative of fraud. The proposed approach is tested on real-world credit card transaction datasets, and the results demonstrate that the GA-based model outperforms traditional methods in terms of detection accuracy, precision, and recall. By evolving better classifiers and reducing the impact of irrelevant features, the Genetic Algorithm offers a robust solution to address the challenges of fraud detection in credit card payment systems. This research highlights the potential of evolutionary algorithms in improving fraud detection models, contributing to the security and reliability of financial transactions.

CHAPTER 3: SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

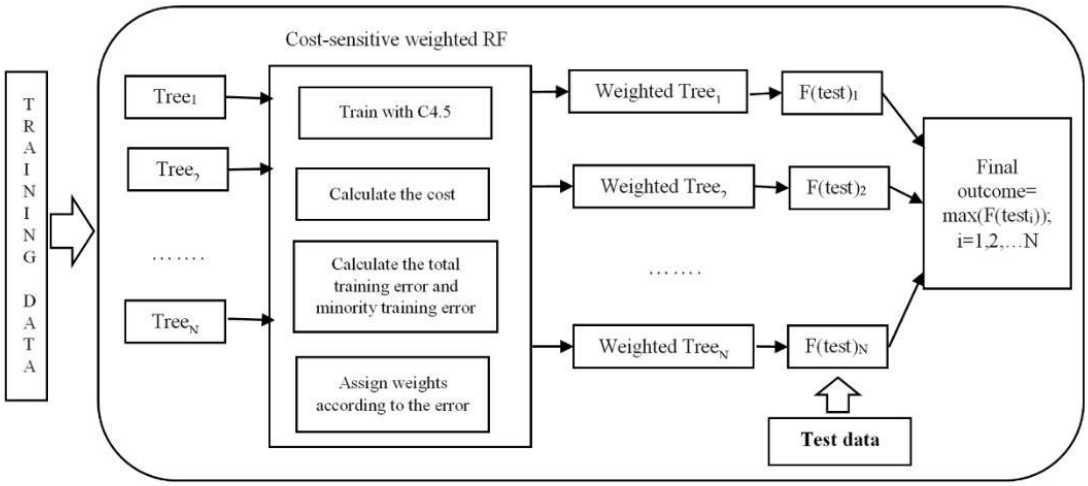


Figure No. 3.1 System Architecture

3.2 MODULES

1)Upload Credit Card Dataset: To collect credit card dataset from available location for system training.

2)Generate Train & Test Model: When we give input to the machine can slit the data like train and test based on 80/20 ratio.

3)Run Random Forest Algorithm: Run this supervised learning algorithm for dataset training or analyze the data inside the system.

4)Detect Fraud from Test Data: This is for prediction the result based on the training.

5)Clean & Fraud Transaction Detection Graph: This is for get the statistical information like bar chart.

3.3 UML DIAGRAMS:

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

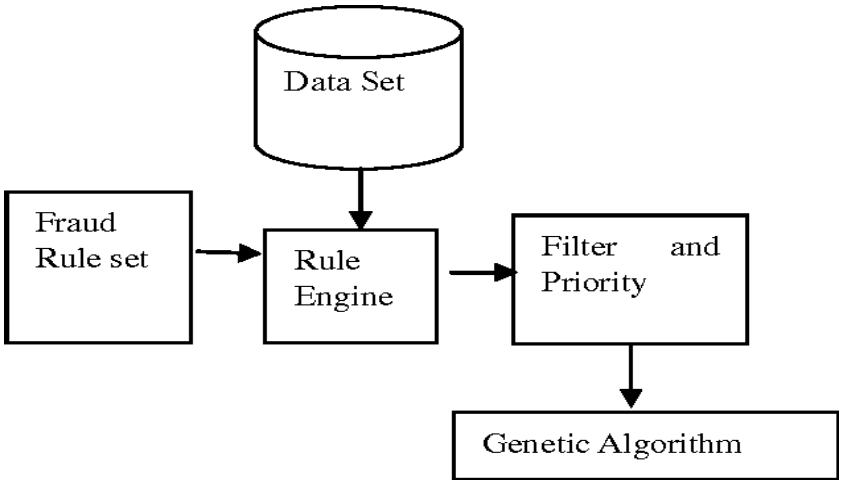


Figure 3.2: Data Flow Diagram

GOALS:

The Primary goals in the design of the UML are as follows:

- Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of OO tools market.
- Integrate best practices.

3.3.1 USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

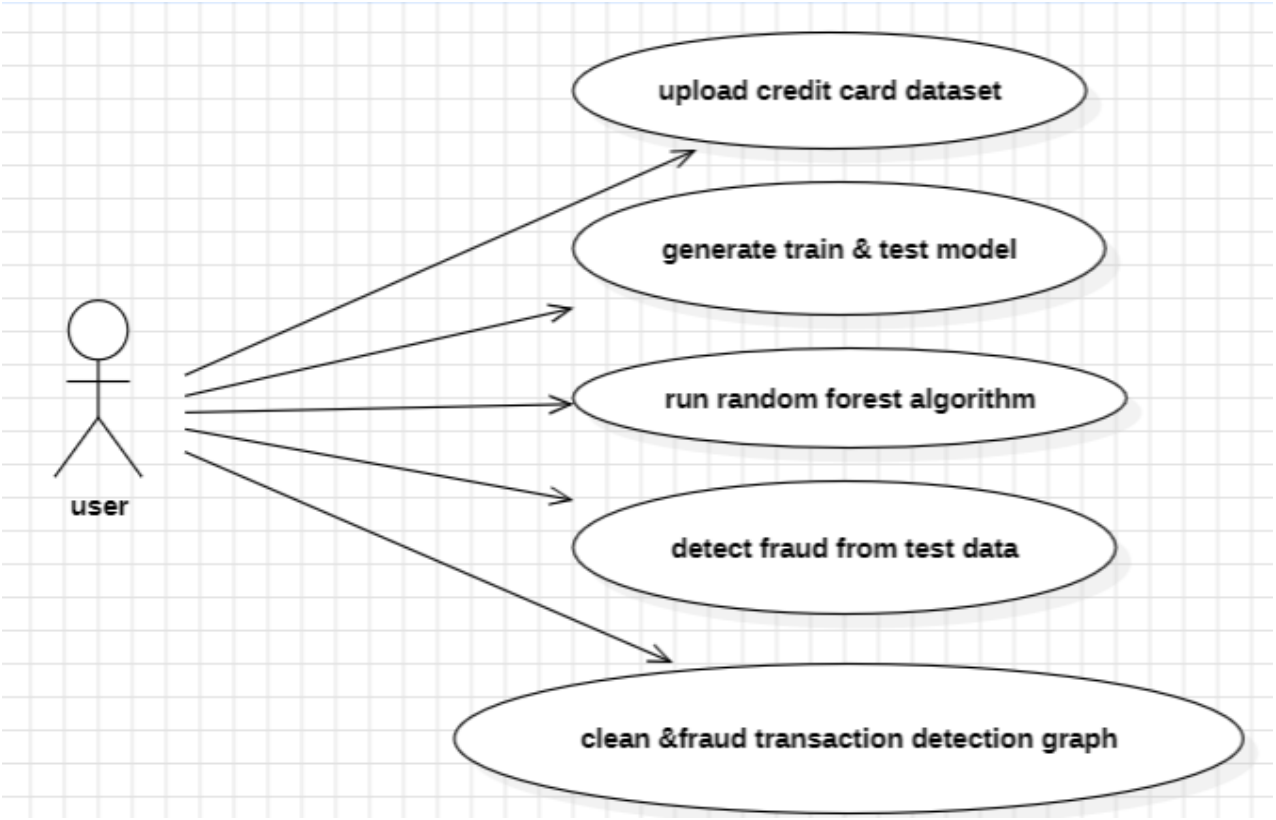


Figure 3.3.1: USE CASE DIAGRAM

3.3.2 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

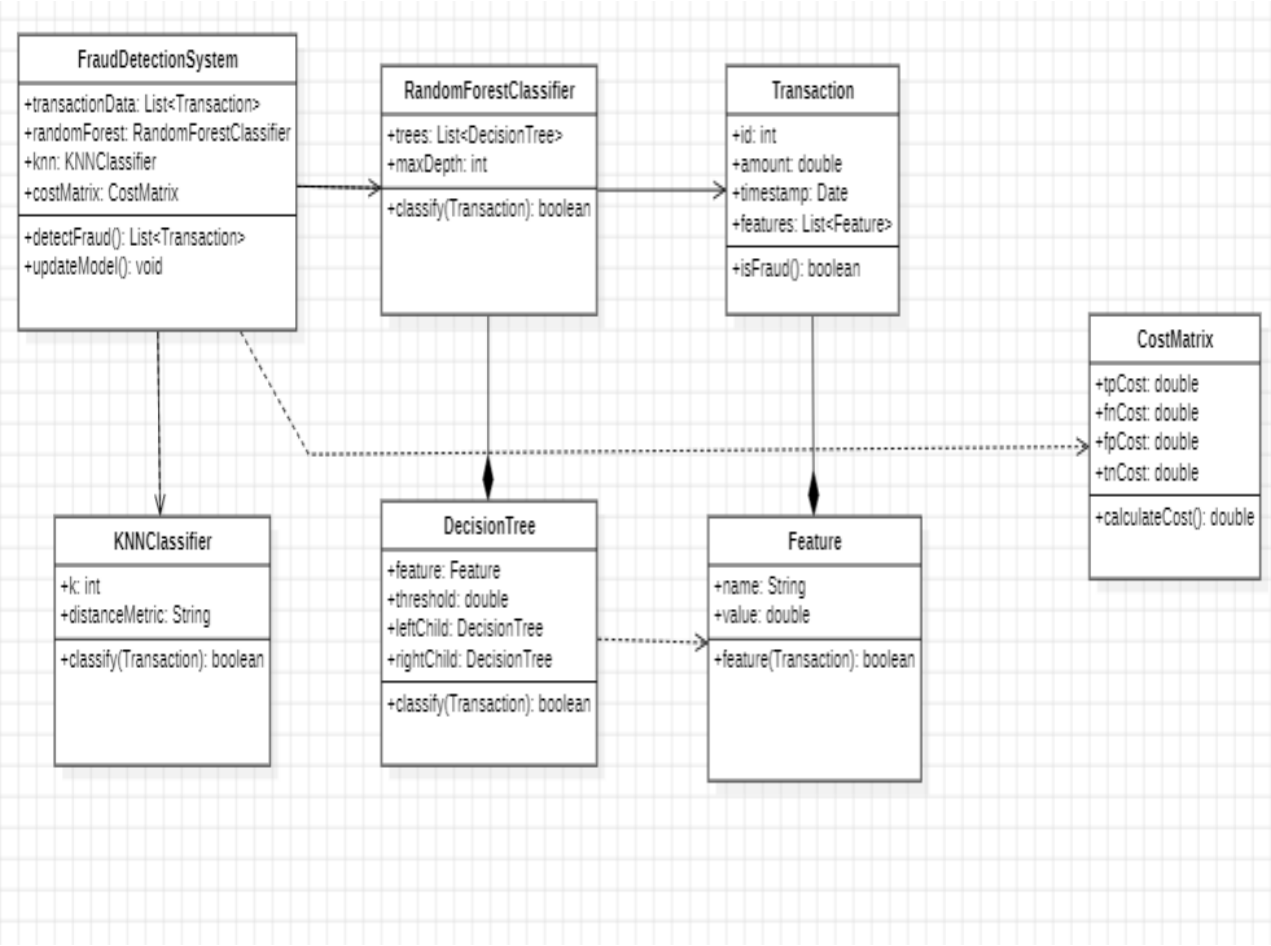


Figure 3.3.2: Class diagram

3.3.3 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

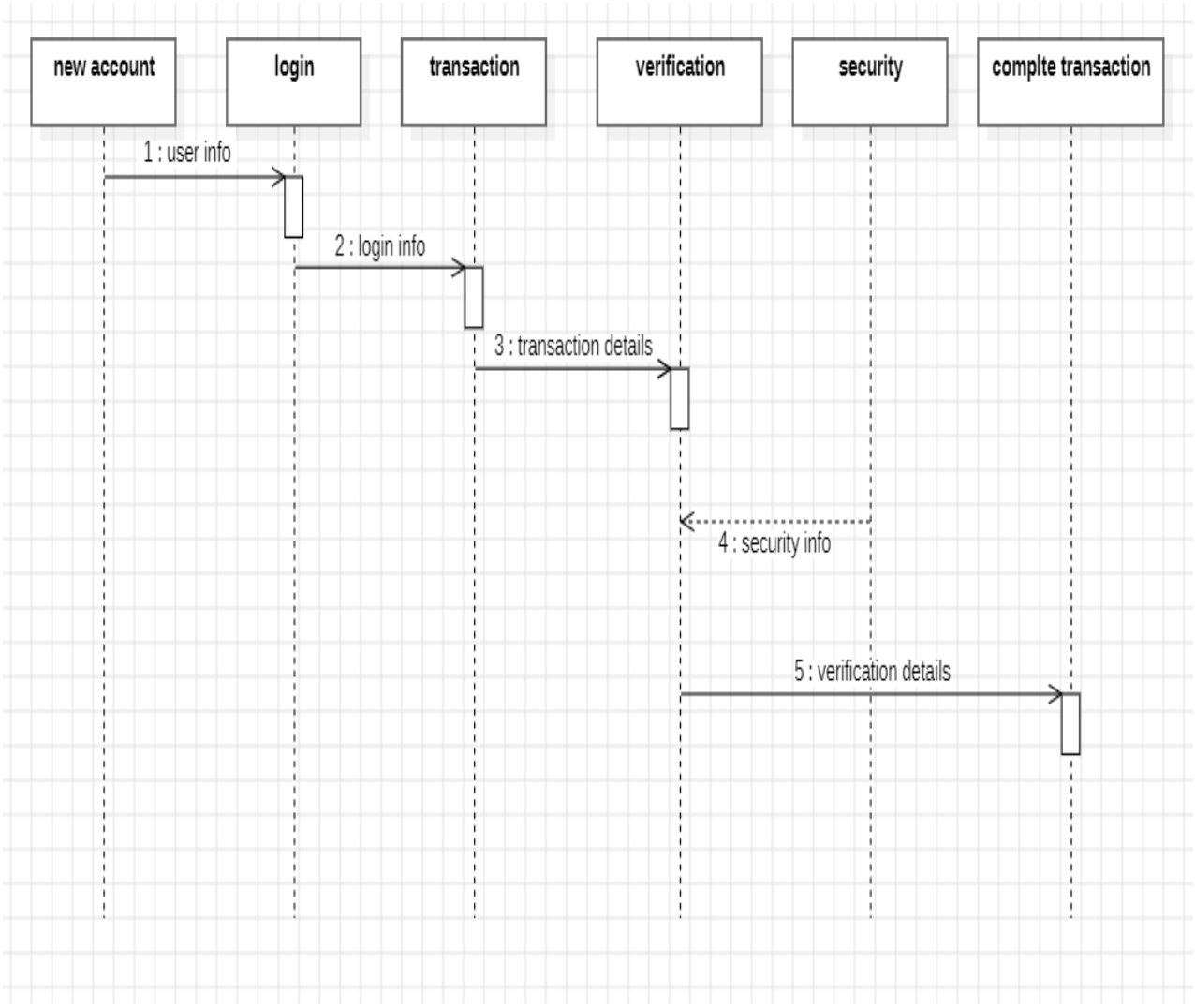


Figure 3.3.3: SEQUENCE DIAGRAM

3.4 SYSTEM REQUIREMENTS

3.4.1 HARDWARE REQUIREMENTS:

- System : intel core i3
- Hard Disk : 2 GB
- Ram : 4 GB

3.4.2 SOFTWARE REQUIREMENTS:

- Operating system : Windows 10
- Coding Language : python
- Tool : PyCharm
- Database : MYSQL.

CHAPTER 4. INPUT DESIGN AND OUTPUT DESIGN

4.1 INPUT DESIGN:

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES:

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2.It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus, the objective of input design is to create an input layout that is easy to follow.

4.2 OUTPUT DESIGN:

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
2. Select methods for presenting information.
3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

CHAPTER 5 : SOFTWARE ENVIRONMENT

5.1 Python:

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. Python features a dynamic type system and automatic memory management.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard libraries which can be used for the following –

- Machine Learning
- GUI Applications (like Kivy , Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv, Pillow)
- Web scraping (like Scrapy, Beautiful Soup, Selenium)
- Test frameworks
- Multimedia

History of Python: -

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wickenden & Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde Informatica (CWI). I don't know how

well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it. Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers.

Advantages of Python: -

Let us see how Python dominates over other languages.

1.Extensive Libraries

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we do not have to write the complete code for that manually.

2. Extensible

As we have seen earlier, Python can be **extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities** to our code in the other language.

4. Improved Productivity

The language's simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet of Things. This is a way to connect the language with the real world.

6.Simple and Easy

When working with Java, you may have to create a class to print **'Hello World'**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and **code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory. These further aids the readability of the code.

8. Object-Oriented

This language supports both the **procedural and object-oriented** programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

9. Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you **download Python** for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to **code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA).** However, you need to be careful enough not to include any system-dependent features.

11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages. Statements are executed one by one; **debugging is easier** than in compiled languages.

Advantages of Python Over Other Languages

1.Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

1. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used it gives you better community support.

The 2019 GitHub annual survey showed us that Python has overtaken Java in the most popular programming language category.

2. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and **machine learning**, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

What is Machine Learning: -

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain. Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

Applications of Machines Learning: -

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

- Emotion analysis
- Sentiment analysis
- Error detection and prevention
- Weather forecasting and prediction
- Stock market analysis and forecasting
- Speech synthesis
- Speech recognition
- Customer segmentation
- Object recognition
- Fraud detection
- Fraud prevention

Advantages of Machine learning: -

1. Easily identifies trends and patterns -

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

2. No human intervention needed (automation)

With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

3. Continuous Improvement

As ML algorithms gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

5. Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

Python Development Steps: -

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt. sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system. Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked. Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting Unicode. Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

- Print is now a function
- Views and iterators instead of lists
- The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
- There is only one integer type left, i.e. int. long is int as well.
- The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behavior. text Vs. Data Instead of Unicode Vs. 8-bit.

6. Purpose: -

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

7. Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library. Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Modules Used in Project: -

TensorFlow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google. TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

NumPy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using NumPy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and I Python shells, the Jupiter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and gallery. For simple plotting the pilot module provides a MATLAB-like interface, particularly when combined with I Python. For the power user, you have full control of line styles, font properties, axes properties, etc. via an object-oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

Install Python Step-by-Step in Windows and Mac:

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace. The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

5.1 Installation of Python on Windows and Mac:

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

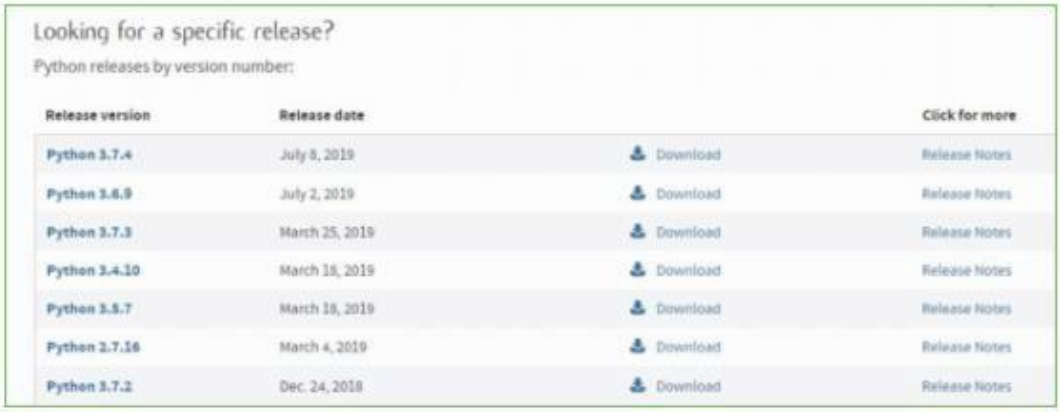
Download the Correct version into the system

Step 1: Go to the official site download and install python using Google Chrome or any other web browser.

Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4



Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.

To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

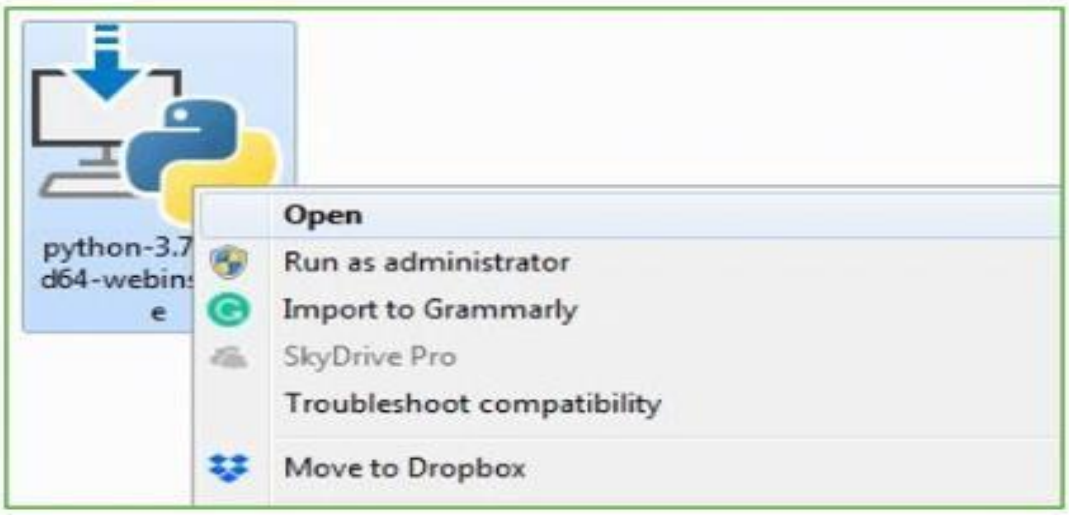
To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process



Step 2: Before you click on Install Now, make sure to put a tick on Add Python 3.7 to PATH.

Step 3: Click on Install NOW After the installation is successful. Click on Close.

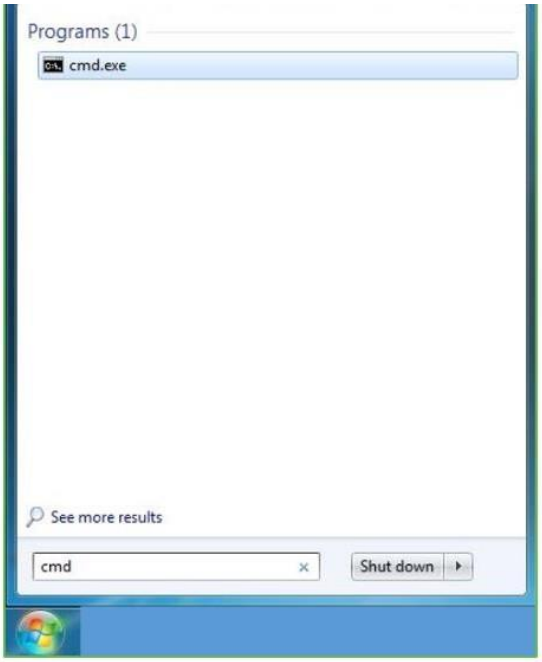
With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

Verify the Python Installation

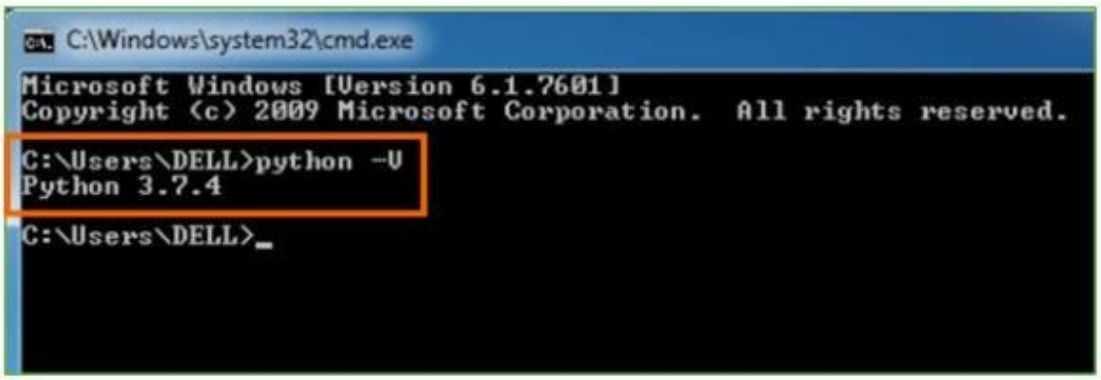
Step 1: Click on Start

Step 2: In the Windows Run Command, type "cmd".



Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type **python –V** and press Enter.



Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

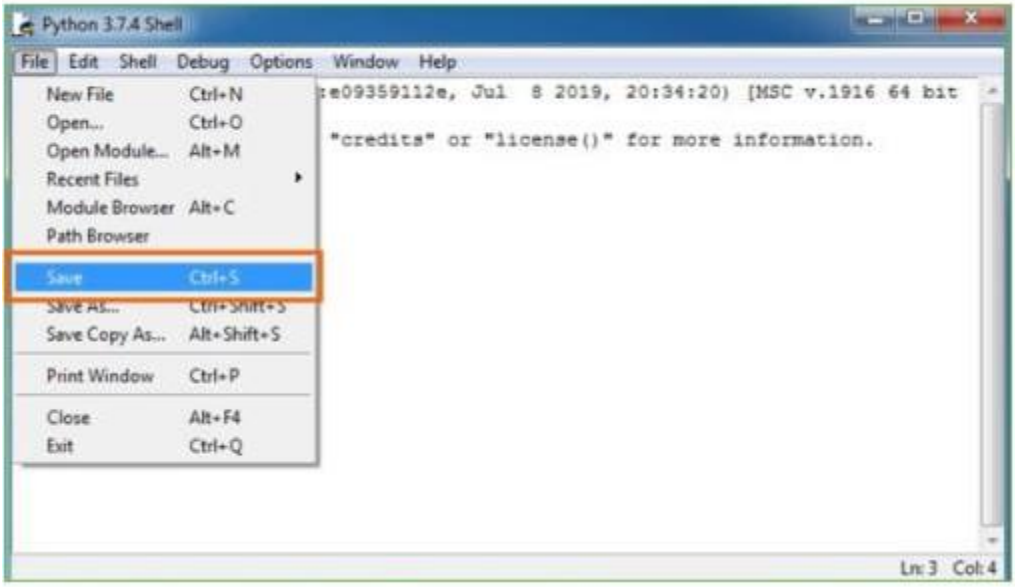
Check how the Python IDLE works

Step 1: Click on Start

Step 2: In the Windows Run command, type "python idle".

Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

Step 4: To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step 6: Now for e.g. **enter print**

CHAPTER 6: SYSTEM STUDY

6.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are:

6.2 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

6.3 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

6.4 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 7: SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

7.1 TYPES OF TESTS:

7.1.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application.it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results. Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

7.1.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components. Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.1.3 Functional test:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centred on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedure: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7.1.4 System Test:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

7.1.5 White Box Testing:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

7.1.6 Black Box Testing:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Test strategy and approach:

Field testing will be performed manually and functional tests will be written in detail.

Test objectives:

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

7.1.7 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

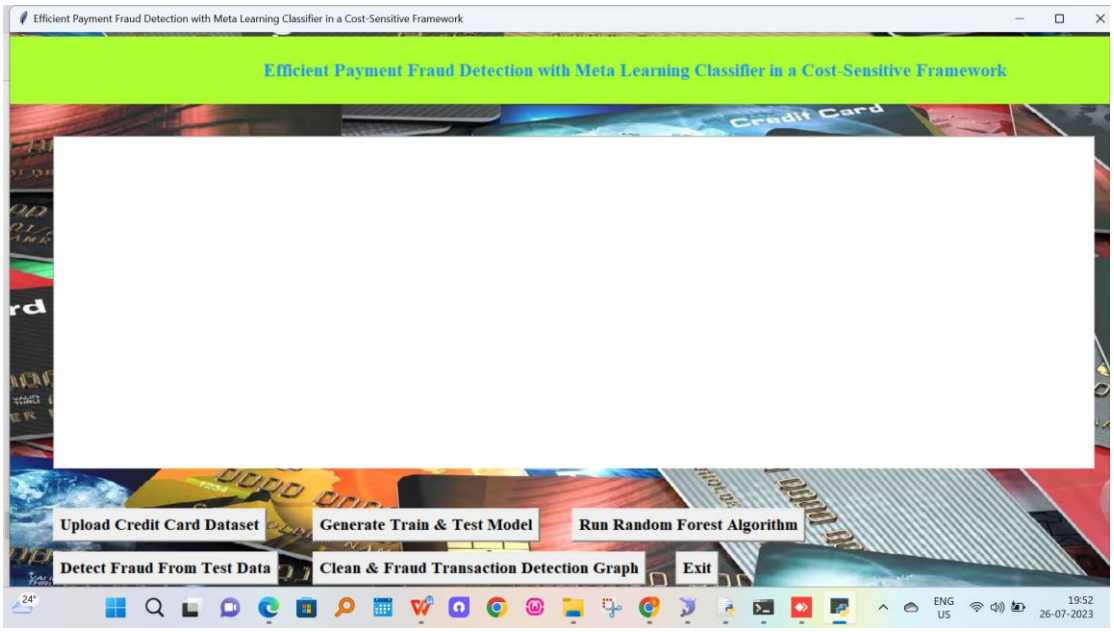
Test Results: All the test cases mentioned above passed successfully. No defects encountered.

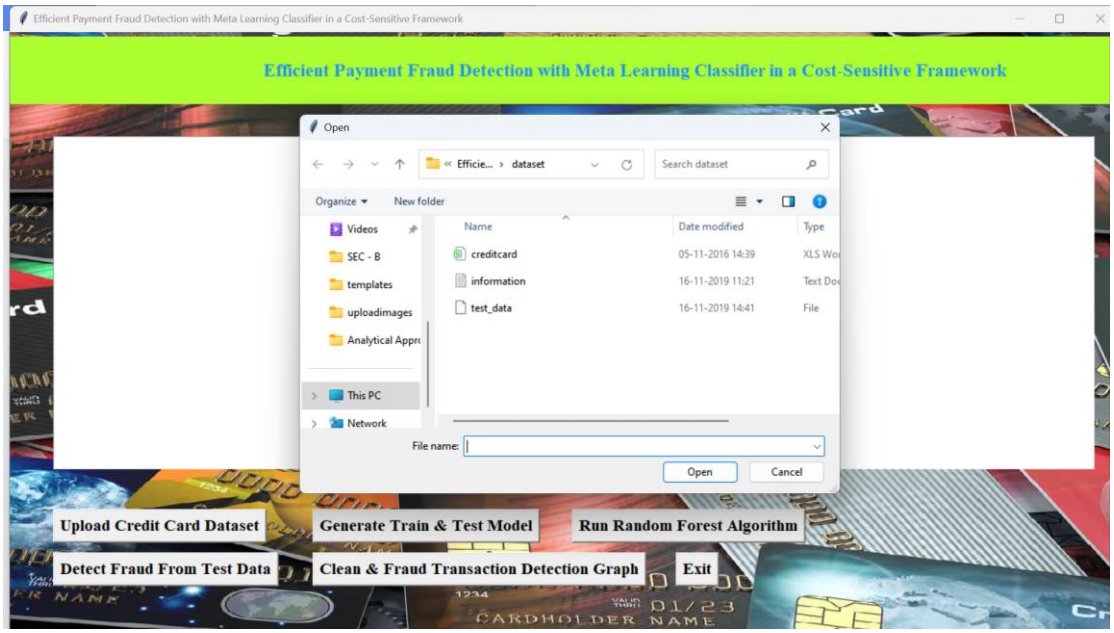
7.2 Test Cases

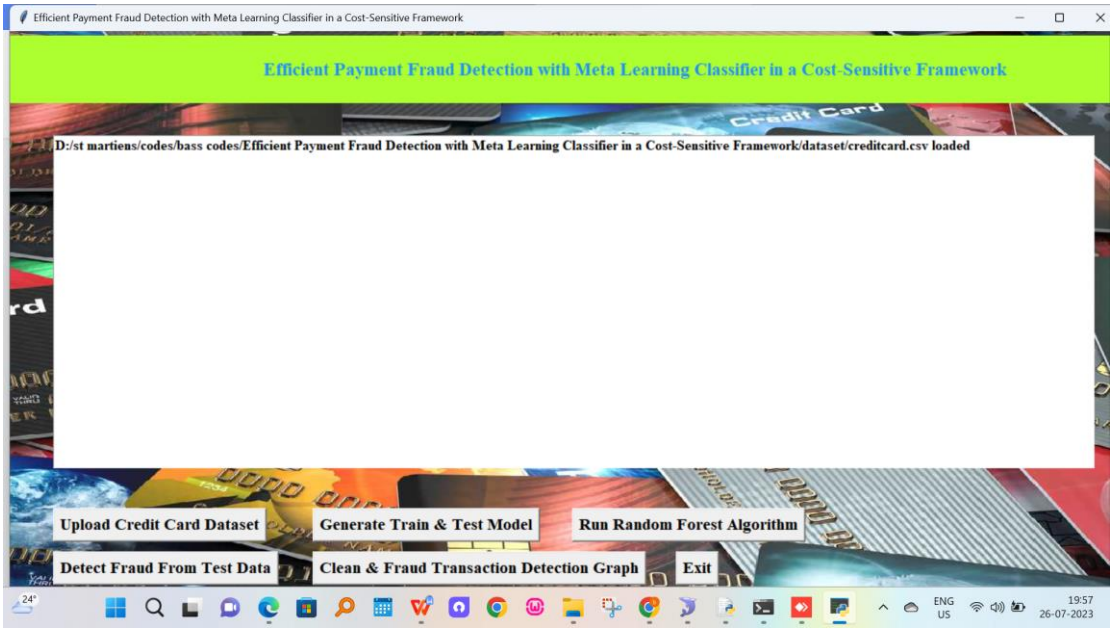
Testcase ID	Description	Input Data	Expected Output	Actual Output	Status
TC_1	Verify model detects fraudulent transactions correctly	Transaction with high-risk features (e.g., high amount, rapid multiple transactions)	Classified as Fraud	Classified as Fraud	Pass
TC_2	Verify model detects genuine transactions correctly	Normal transaction with usual behaviour	Classified as Not Fraud	Classified as Not Fraud	Pass
TC_3	Verify model performance on unbalanced dataset	Dataset with 98% non-fraud, 2% fraud transactions	Higher weight assigned to minority class,	Higher weight assigned, improved fraud detection rate	Pass

TC_4	Check model response to missing values	Transaction with missing features (e.g., no transaction amount)	Model handles missing values without failure	Model handles missing values	Pass
TC_5	Evaluate model on a real-world dataset	Real transaction dataset	Fraud detection rate improves over traditional methods	Model detects more fraud than baseline	Pass
TC_6	Validate model scalability with large datasets	Large dataset with millions of transactions	Model processes efficiently without failure	Model processes successfully	Pass
TC_7	Verify impact of changing cost function	Different cost function applied	Fraud detection rate changes accordingly	Expected changes observed	Pass
TC_8	Evaluate model runtime performance	Measure execution time for fraud detection	Model runs within acceptable time limit	Model executes in expected time	Pass

CHAPTER 8: RESULTS

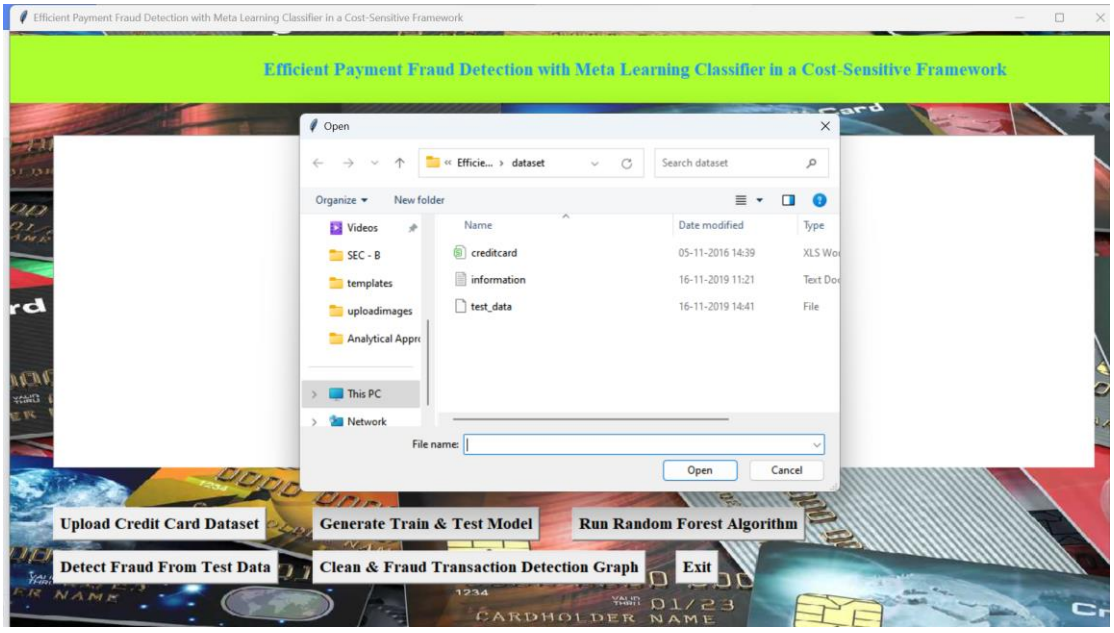




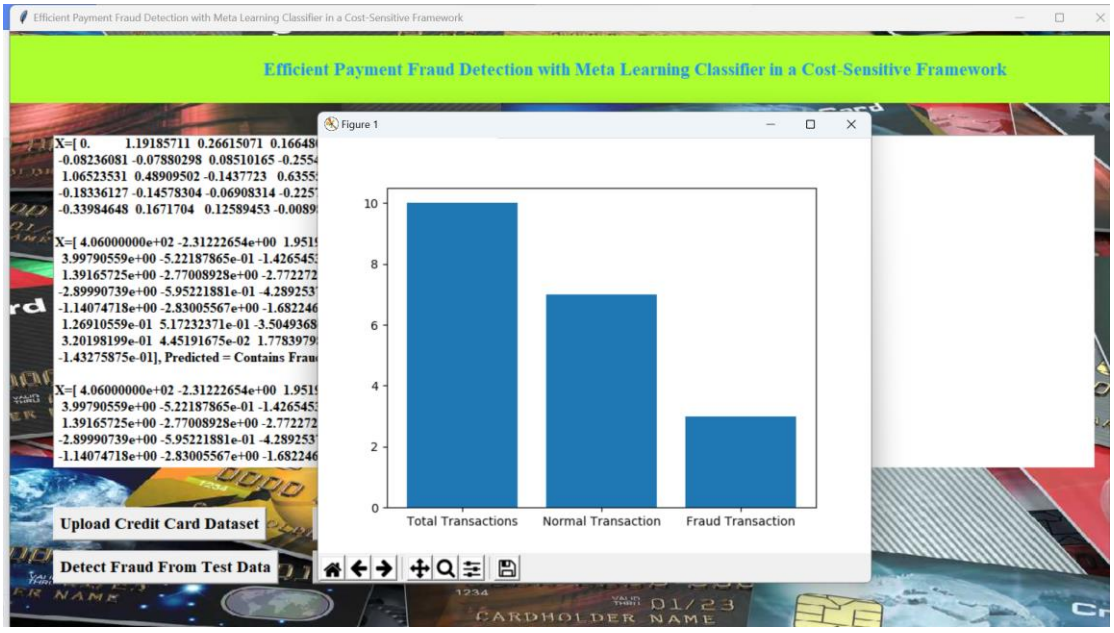

Efficient Payment Fraud Detection with Meta Learning Classifier in a Cost-Sensitive Framework
Efficient Payment Fraud Detection with Meta Learning Classifier in a Cost-Sensitive Framework
D:/st martiens/codes/bass codes/Efficient Payment Fraud Detection with Meta Learning Classifier in a Cost-Sensitive Framework/dataset/creditcard.csv loaded
Upload Credit Card Dataset
Generate Train & Test Model
Run Random Forest Algorithm
Detect Fraud From Test Data
Clean & Fraud Transaction Detection Graph
Exit


Efficient Payment Fraud Detection with Meta Learning Classifier in a Cost-Sensitive Framework
Efficient Payment Fraud Detection with Meta Learning Classifier in a Cost-Sensitive Framework
D:/st martiens/codes/bass codes/Efficient Payment Fraud Detection with Meta Learning Classifier in a Cost-Sensitive Framework/dataset/creditcard.csv loaded
Train & Test Model Generated
Total Dataset Size : 284807
Split Training Size : 199364
Split Test Size : 85443
Upload Credit Card Dataset
Generate Train & Test Model
Run Random Forest Algorithm
Detect Fraud From Test Data
Clean & Fraud Transaction Detection Graph
Exit


Efficient Payment Fraud Detection with Meta Learning Classifier in a Cost-Sensitive Framework
Efficient Payment Fraud Detection with Meta Learning Classifier in a Cost-Sensitive Framework
D:/st martiens/codes/bass codes/Efficient Payment Fraud Detection with Meta Learning Classifier in a Cost-Sensitive Framework/dataset/creditcard.csv loaded
Train & Test Model Generated
Total Dataset Size : 284807
Split Training Size : 199364
Split Test Size : 85443
Prediction Results
Random Forest Accuracy
Accuracy : 99.78816286881312
Upload Credit Card Dataset
Generate Train & Test Model
Run Random Forest Algorithm
Detect Fraud From Test Data
Clean & Fraud Transaction Detection Graph
Exit


Efficient Payment Fraud Detection with Meta Learning Classifier in a Cost-Sensitive Framework
Efficient Payment Fraud Detection with Meta Learning Classifier in a Cost-Sensitive Framework
Open
Effic... › dataset
Search dataset
Organize
New folder
Videos
SEC - B
templates
uploadimages
Analytical Appr
This PC
Network
Name
Date modified
Type
creditcard
05-11-2016 14:39
information
16-11-2019 11:21
test_data
16-11-2019 14:41
File name:
Open
Cancel
Upload Credit Card Dataset
Generate Train & Test Model
Run Random Forest Algorithm
Detect Fraud From Test Data
Clean & Fraud Transaction Detection Graph
Exit


Efficient Payment Fraud Detection with Meta Learning Classifier in a Cost-Sensitive Framework
Efficient Payment Fraud Detection with Meta Learning Classifier in a Cost-Sensitive Framework
X=[0. 1.19185711 0.26615071 0.16648011 0.44815408 0.06001765
-0.08236081 -0.07880298 0.08510165 -0.25542513 -0.16697441 1.61272666
1.06523531 0.48909502 -0.1437723 0.63555809 0.46391704 -0.11480466
-0.18336127 -0.14578304 -0.06908314 -0.22577525 -0.63867195 0.10128802
-0.33984648 0.1671704 0.12589453 -0.0089831 0.01472417], Predicted = Transaction Contains Cleaned Signatures
X=[4.06000000e+02 -2.31222654e+00 1.95199201e+00 -1.60985073e+00
3.99790559e+00 -5.22187865e-01 -1.42654532e+00 -2.53738731e+00
1.39165725e+00 -2.77008928e+00 -2.77227214e+00 3.20203321e+00
-2.89990739e+00 -5.95221881e-01 -4.28925378e+00 3.89724120e-01
-1.14074718e+00 -2.83005567e+00 -1.68224682e-02 4.16955705e-01
1.26910559e-01 5.17232371e-01 -3.50493686e-02 -4.65211076e-01
3.20198199e-01 4.45191675e-02 1.77839798e-01 2.61145003e-01
-1.43275875e-01], Predicted = Contains Fraud Transaction Signature So Transaction Stoped
X=[4.06000000e+02 -2.31222654e+00 1.95199201e+00 -1.60985073e+00
3.99790559e+00 -5.22187865e-01 -1.42654532e+00 -2.53738731e+00
1.39165725e+00 -2.77008928e+00 -2.77227214e+00 3.20203321e+00
-2.89990739e+00 -5.95221881e-01 -4.28925378e+00 3.89724120e-01
-1.14074718e+00 -2.83005567e+00 -1.68224682e-02 4.16955705e-01
Upload Credit Card Dataset
Generate Train & Test Model
Run Random Forest Algorithm
Detect Fraud From Test Data
Clean & Fraud Transaction Detection Graph
Exit


Efficient Payment Fraud Detection with Meta Learning Classifier in a Cost-Sensitive Framework
Efficient Payment Fraud Detection with Meta Learning Classifier in a Cost-Sensitive Framework
Figure 1
10
8
6
4
2
0
Total Transactions
Normal Transaction
Fraud Transaction
Upload Credit Card Dataset
Detect Fraud From Test Data

CHAPTER 9: CONCLUSION

9.1 Conclusion

A cost-sensitive random forest-based ensemble learning technique has been proposed for effective detection of credit card fraud. The imbalanced nature of credit card data has been investigated in this work. A misclassification ratio-based cost-function has been integrated into the error-formulation of the generated sub-tress in RF-bagging. The cost-function facilitates to determine the predictive ability of sub-tree, so that the sub-tree with highest predictive ability can have maximum weight age. The final outcome of the test-set is determined based on the outcome, yielded by the maximally weighted tree. The experimental results achieved have manifested the efficiency of the proposed model in effective handling of imbalanced cases in case of credit card fraud detection. The proposed model has not been validated for high-dimensional datasets. The proposed model can be extended by integrating with different data cleaning techniques such as sampling or feature selection (or extraction) algorithm to be implemented in high-dimensional datasets. The imbalanced nature of the detection methods and its effects over the performance has not been investigated in this paper. A deep exploration of the issue in accordance with the computational performance of the credit card fraud detection techniques is another scope of future study.

9.2 Future Enhancement

For future enhancement of the proposed cost-sensitive weighted random forest algorithm for credit card fraud detection, several directions can be explored to further improve its effectiveness and adaptability. Firstly, the integration of advanced data balancing techniques such as SMOTE (Synthetic Minority Over-sampling Technique) or ADASYN (Adaptive Synthetic Sampling) in conjunction with the cost-sensitive approach could enhance model performance by providing a more representative training set. Secondly, the model could be extended to incorporate temporal and behavioral features of users, leveraging time-series analysis and user profiling to detect sophisticated and evolving fraud patterns. Additionally, implementing ensemble methods that combine multiple machine learning algorithms, including deep learning models such as LSTM or autoencoders, may boost detection accuracy, especially for complex fraud scenarios. Real-time fraud detection capabilities can also be introduced to enable immediate response to suspicious activities. Moreover, continuous learning frameworks could be applied to allow the model to adapt to new patterns over time, maintaining robustness against concept drift. Finally, deploying the system in real-world banking environments with feedback loops could improve its reliability and offer opportunities for fine-tuning based on actual transaction data and expert inputs.

CHAPTER 10: BIBLOGRAPHY

[1] L. Breiman, Random forests. Machine Learning, vol. 45, issue 1, pp: 5-32, 2021.

[2] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, Credit card fraud detection using Machine Learning Techniques: A Comparative Analysis, pp: 1-9, 2020.

[3] S. Patil, H. Somavanshi, J. Gaikwad, A. Deshmane, and R. Badgujar, Credit Card Fraud Detection Using Decision Tree Induction Algorithm, International Journal of Computer Science and Mobile Computing (IJCSMC), vol.4, issue 4, pp. 92-95, 2015. ISSN: 2320-088X

[4] G. Singh, R. Gupta, A. Rastogi, M. D. S. Chandel, and A. Riyaz, A Machine Learning Approach for Detection of Fraud based on SVM, International Journal of Scientific Engineering and Technology, vol.1, issue 3, pp. 194-198, 2021.

[5] A. C. Bahnsen, A. Stojanovic, D. Aouada, and B. Ottersten, Improving credit card fraud detection with calibrated probabilities. In Proceedings of the 2020 SIAM International Conference on Data Mining, pp. 677-685, 2021.

[6] F. N. Ogwueleka, Data Mining Application in Credit Card Fraud Detection System, Journal of Engineering Science and Technology, vol. 6, issue 3, pp. 311 – 322, 2020.

[7] K. RamaKalyani, and D. UmaDevi, Fraud Detection of Credit Card Payment System by Genetic Algorithm, International Journal of Scientific & Engineering Research, vol. 3, issue 7, pp. 1 – 6, 2021.

[8] P. L., Meshram, and P. Bhanarkar, Credit and ATM Card Fraud Detection Using Genetic Approach, International Journal of Engineering Research & Technology (IJERT), vol. 1, issue 10, pp. 1 – 5, 2019.

[9] K. R. Seeja, and M. Zareapoor, FraudMiner: A Novel Credit Card Fraud Detection Model Based on Frequent Itemset Mining, The Scientific World Journal, Hindawi Publishing Corporation, vol. 2018,

[10] M. Zareappor, P. Shamsolmoali, Application of Credit card Fraud Detection: Based onBagging Ensemble Classifier, International Conference on Intelliegent Computing, Communication, & Convergence (ICCC-2016), In Procedia Computer Science, Vol. 48, pp: 679 – 685, 2016.

[11] V, Patil, U. K. Lilhore, A Survey on Different Data Mining & Machine Learning Methods for Credit Card Fraud Detection, International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Vol. 3, Issue 5, pp:320-325, 2016.

[12] S. Stolfo, D. W. Fan, W. Lee, A. Prodromidis, and P. Chan, Credit card fraud detection using meta-learning: Issues and initial results. In AAAI-97 Workshop on Fraud Detection and Risk Management., 2014.

[13] A. Y. Ng, and M. I. Jordan, On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. Advances in neural information processing systems, Vol. 2, pp: 841-848, 2015.

[14] C. Phua, D. Alahakoon, and V. Lee, Minority report in fraud detection: classification of skewed data. ACM sigkdd explorations newsletter, vol. 6, issue 1, pp: 50-59, 2014.

[15] F. Chu, Y. Wang, C. Zaniolo, An adaptive learnin approach for noisy data streams. IEEE International Conference on Data Mining, 2004 (ICDM'04), pp: 351- 354, 2014.

[16] A. Shen, R., Tong, and Y. Deng, Application of classification models on credit card fraud detection. IEEE International Conference on Service Systems and Service Management, pp: 1-4, 2015.

[17] Y. Sahin, and, E. Duman, Detecting Credit Card Fraud by Decision Trees and Support Vector Machines, Proceeding of International Multi-Conference of Engineers and Computer Scientists (IMECS 2012), vol. 1, pp: 1- 6, ISBN: 978-988-18210-3-4, ISSN: 2078-0966 (Online), 2011.

[18] Y. Sahin, and, E. Duman, Detecting credit card fraud by ANN and logistic regression. In IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA), pp: 315-319, 2011.

has been employed for determining the final outcome of the test-set. Any kind of pre-processing of the data (data cleaning, feature selection/ extraction) is not employed in the proposed scheme. The objective of the proposed work is to define an improved and effective RF algorithm, for credit card fraud detection, in presence of imbalanced cases.

YUKTHI INNOVENTION CERTIFICATE

MoE's INNOVATION CELL (GOVERNMENT OF INDIA)

INSTITUTION'S INNOVATION COUNCIL
MOE'S INNOVATION CELL

INSTITUTION'S INNOVATION COUNCIL (Ministry of Education Initiative)

Institute Name:
Malla Reddy Institute of Technology & Science

Title of the Innovation/Prototype:
Efficient Payment Fraud Detection with Meta Learning Classifier in A Cost-Sensitive Framework

Team Lead Name:	Team Lead Email:	Team Lead Phone:	Team Lead Gender:
Rapelly Sahith	sahithrapelly2579@gmail.com	9553993878	Male
FY of Development:	**Developed as part of:**	**Innovation Type:**	**TRL LEVEL:**
2024-25	Academic Requirement/Study Project	Process,Service	4

MRL Level:
MRL 3: Manufacturing proof of concept developed

IRL Level:
IRL 4: Prototype Low-Fidelity Minimum Viable Product (MVP): "Low-fidelity" - A representative of the component or system that has limited ability to provide anything but initial information about the end product.

Theme:
Defence & Security,Fashion and Textiles,

Define the problem and its relevance to today's market / society / industry need:
Credit card fraud is a growing concern in today's digital economy, leading to substantial financial losses for individuals and institutions. Traditional fraud detection systems struggle with highly imbalanced datasets, high false positive rates and adaptability to evolving fraud techniques. With the rapid increase in online transactions, there is an urgent need for accurate, cost-sensitive, and scalable fraud detection systems. This project addresses the challenge by proposing a meta-learning-based, cost-sensitive random forest algorithm to enhance detection of minority class (fraud) instances, offering a robust solution aligned with the current needs of financial industries and e-commerce platforms.

Describe the Solution / Proposed / Developed:
The proposed solution is a cost-sensitive weighted random forest algorithm enhanced with meta-learning techniques for efficient credit card fraud detection. It addresses the issue of imbalanced datasets by assigning higher weights to minority class instances during training. Each tree in the ensemble is evaluated based on its predictive ability for fraudulent transactions, and final decisions are made using a weighted voting mechanism. This improves accuracy, reduces false positives, and adapts to evolving fraud patterns. The system is designed to be computationally efficient and scalable, making it suitable for real-time fraud detection in high-volume transaction environments.

Explain the uniqueness and distinctive features of the (product / process / service) solution:
The solution uniquely integrates a cost-sensitive learning framework with a meta-learning-based weighted random forest, specifically designed to handle imbalanced credit card fraud datasets. custom cost function during training. Each decision tree is evaluated and weighted based on its accuracy in detecting fraudulent transactions, improving overall ensemble performance. The model replaces standard majority voting with an error-based weighting mechanism, enhancing precision and reducing false positives. Its adaptability, scalability, and real-time efficiency distinguish it as a powerful tool for dynamic fraud detection in financial and digital transaction systems.

How your proposed / developed (product / process / service) solution is different from similiar kind of product by the competitors if any:
Unlike conventional fraud detection systems that rely on standard machine learning algorithms and majority voting, our solution introduces a cost-sensitive weighted random forest enhanced with meta-learning. Competing models often fail to effectively handle imbalanced datasets, leading to high false negatives or positives. Our approach uniquely assigns dynamic weights to each tree based on its fraud detection performance which most existing solutions lack. This makes our model more adaptive, accurate, and suitable for real-time, large-scale transaction environments compared to current industry-standard techniques.

Is there any IP or Patentable Component associated with the Solution?:
No

Has the Solution Received any Innovation Grant/Seefund Support?:
No

Are there any Recognitions (National/International) Obtained by the Solution?:
No

***Is the Solution Commercialized either through Technology Transfer or Enterprise Development/Startup?:**
No

Had the Solution Received any Pre-Incubation/Incubation Support?:
No

Video URL:
https://drive.google.com/file/d/16GuBQ8UEjt3naAi83TBGec2QnqE5BWta/view?usp=sharing

Innovation Photograph:
View File

Downloaded on: 21-04-2025

This report is electronically generated against Yukti - National Innovation Repository Portal.