

A PROJECT REPORT ON
FINANCIAL FRAUD DETECTION USING MACHINE
LEARNING

Submitted in partial fulfillment for the award of the degree of
BACHELOR OF TECHNOLOGY

In
Computer Science and Engineering

By
S.Madhuri (21A81A0553)
Y.Revathi (21A81A0564)
S.S.D.Lavanya (21A81A0551)
M.Kartheek(22A85A0506)
A.Chandra Naga Sai (21A81A0502)

Under the Esteemed Supervision of
Mrs.N.Hiranmayee, M.Tech.,
Sr.Asst.Professor



Department of Computer Science and Engineering(Accredited by N.B.A.)
SRI VASAVI ENGINEERING COLLEGE(Autonomous)
(Affiliated to JNTUK, Kakinada)
Pedatadepalli, Tadepalligudem-534101, A.P 2023-2024

SRI VASAVI ENGINEERING COLLEGE (Autonomous)

Department Of Computer Science and Engineering

Pedatadepalli, Tadepalligudem



Certificate

This is to certify that the Project Report entitled “**Financial Fraud Detection Using Machine Learning**” submitted by **S.MADHURI(21A81A0553), Y.REVATHI(21A81A0564), S.S.D.LAVANYA(21A81A0551), M.KARTHEEEK (22A85A0506), A.CHANDRA NAGA SAI(21A81A0502)** for the award of the degree of Bachelor of Technology in the Department of Computer Science and Engineering during the academic year 2023-2024.

Name of Project Guide

Mrs. N. Hiranmayee, M.Tech.,
Sr.Asst.Professor

Head of the Department

Dr. D. Jaya Kumari, M.Tech.,Ph.D..
Professor & HOD.

External Examiner

DECLARATION

We hereby declare that the project report entitled “**Financial Fraud Detection Using Machine Learning**” submitted by us to Sri Vasavi Engineering College(Autonomous), Tadepalligudem, affiliated to JNTUK Kakinada in partial fulfillment of the requirement for the award of the degree of B.Tech in Computer Science and Engineering is a record of Bonafide project work carried out by us under the guidance of **Mrs.N.Hiranmayee**,_{M.Tech.} We further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree in this institute or any other institute or University.

Project Associates

S.Madhuri (21A81A0553)

Y.Revathi (21A81A0564)

S.S.D.Lavanya(21A81A0551)

M.Kartheek(22A85A0506)

A.Chandra Naga Sai(21A81A0502)

ACKNOWLEDGEMENT

First and foremost, we sincerely salute to our esteemed institute **SRI VASAVI ENGINEERING COLLEGE**, for giving us this golden opportunity to fulfill our warm dream to become an engineer. Our sincere gratitude to our project guide **Mrs. N. Hiranmayee**, M.Tech. Department of Computer Science and Engineering, for her timely cooperation and valuable suggestions while carrying out this project.

We express our sincere thanks and heartfelt gratitude to **Dr. D. Jaya Kumari**, Professor & Head of the Department of Computer Science and Engineering, for permitting us to do our project.

We express our sincere thanks and heartfelt gratitude to **Dr. G.V.N.S.R. Ratnakara Rao**, Principal, for providing a favourable environment and supporting us during the development of this project.

Our special thanks to the management and all the teaching and non-teaching staff members, Department of Computer Science and Engineering, for their support and cooperation in various ways during our project work. It is our pleasure to acknowledge the help of all those respected individuals.

We would like to express our gratitude to our parents, friends who helped to complete this project.

Project Associates

S.Madhuri(21A81A0553)

Y.Revathi (21A81A0564)

S.S.D.Lavanya(21A81A0551)

M.Kartheek(22A85A0506)

A.Chandra Naga Sai(21A81A0502)

ABSTRACT

The Project Financial Fraud Detection gives basic ideas about detection of fraudulent activities in finance. Nowadays AI has found its usage and appreciation in many fields from medical sector to army & military, it has found its application in marketing and digital sector to track the spending habits of people who are online, and so far, it has no signs of stopping now. This report thoroughly details how machine learning algorithms are used in fraud detection, it's pros and limitations.

From the day when payment systems emerged, there have been people willing to find novel ways to access someone's finances illegally. These menacing hazards has grown in the current period, as the majority of transactions are now completed entirely online. Using Machine learning algorithms to detect fraud is a process in which the data is investigated through various techniques to achieve the best possible outcomes in detecting and impeding fraudulent transactions.

With the ever-increasing amount of data generated by digital transactions, manual fraud detection processes have become inefficient and time-consuming. AI, on the other hand, can analyse large volumes of data in real-time and identify patterns that may indicate fraudulent activity. Overall, AI technologies has revolutionized the field of fraud detection and is poised to play an increasingly important role in keeping digital transactions safe and secure in the years to come.

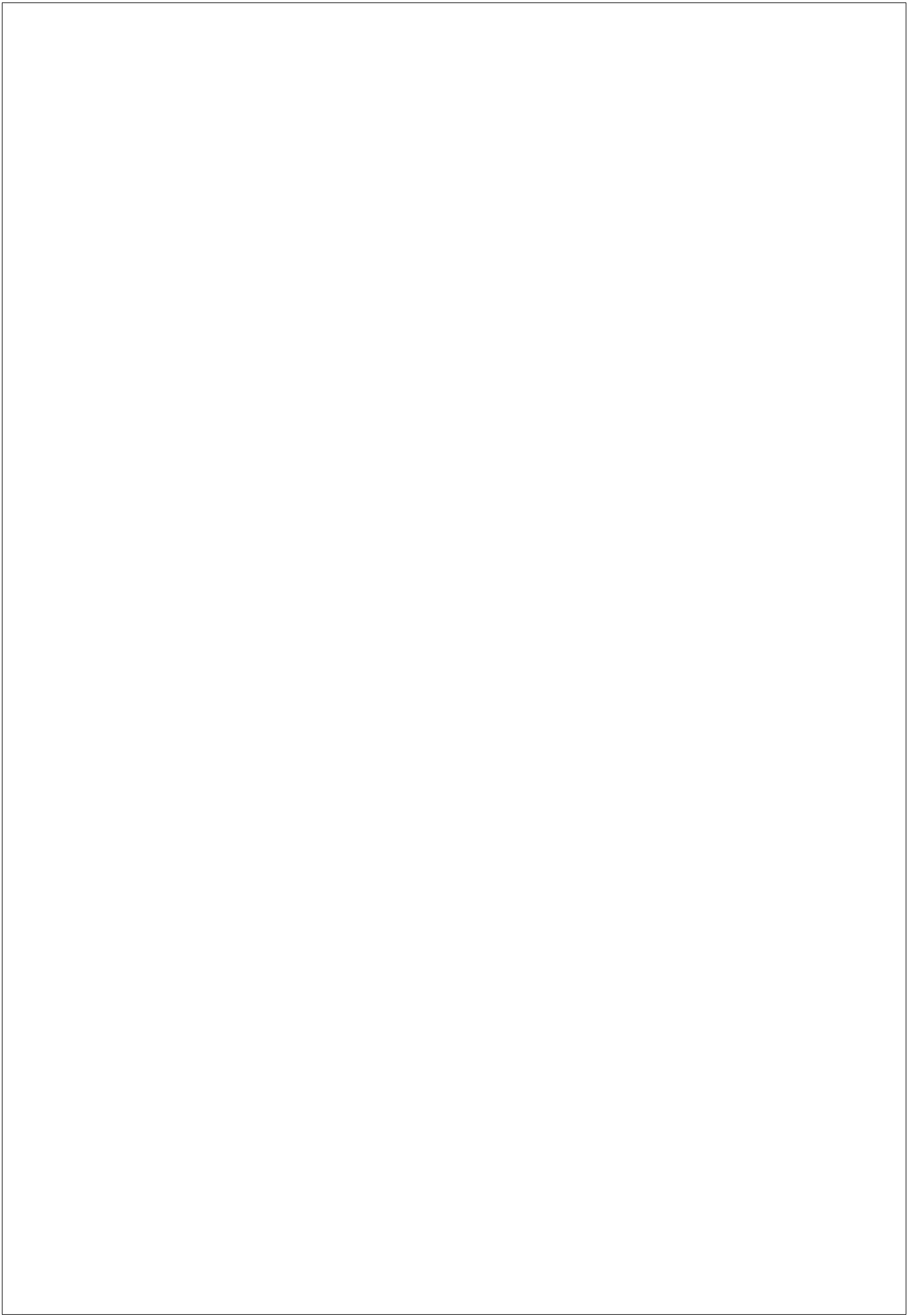
TABLE OF CONTENTS

S.NO	TITLE	PAGE NO
Chapter 1	INTRODUCTION	1-3
1.1	Introduction	2
1.2	Motivation	2
1.3	Objective	2
1.4	Scope	2
1.5	Project Outline	3
Chapter 2	LITERATURE SURVEY	4-5
Chapter 3	SYSTEM STUDY AND ANALYSIS	6-13
3.1	Problem Statement	7
3.2	Existing System	7
3.4	Proposed System	8
3.4.1	Random Forest	9
3.5	Advantages of Proposed System	11
3.6	Functional Requirements	11
3.7	Non-Functional Requirements	12
3.8	Hardware Requirements	13
3.9	Software requirements	13
Chapter 4	SYSTEM DESIGN	14-16
4.1	System Architecture	15
Chapter 5	TECHNOLOGIES	17-20
5.1	Python	18
5.2	About Python	19
5.3	Necessary libraries of Python	19
5.4	Dependencies-VS Code	20
Chapter 6	IMPLEMENTATION	21-24
6.1	Implementation	22
Chapter 7	TESTING	25-28

7.1	Purpose of testing	26
7.2	Types of testing	26
7.3	Test Cases	28
Chapter 8	OUTPUT SCREENSHOTS	29-31
Chapter 9	CONCLUSION AND FUTURE WORK	32-33
9.1	Conclusion	33
9.2	Future Work	33
9.3	References	33

LIST OF FIGURES

FIG NO	TITLE	PAGE NO
Fig.1	Proposed System	5
Fig.2	System Architecture	15
Fig.3	Countplot of class variable	23
Fig.4	Metric Results	24
Fig.5	Output analysis	30
Fig.6	Classification Report	31
Fig.7	ROC Curve	31



CHAPTER - 1

INTRODUCTION

1.1 Introduction:

We are living in a world which is rapidly adopting digital payments systems. In the fast-paced world of finance, where countless transactions occur every second, ensuring the safety of financial systems and protecting hard-earned money is of utmost importance. This is where the power of machine learning steps in to act as a vigilant guardian against fraudulent activities. An effective fraud detection system should be able to detect fraudulent transactions with high accuracy and efficiency. Designing an accurate and efficient fraud detection system that is low on false positives but detects fraudulent activity effectively is a significant challenge for researchers. Fraud is a problematic issue in many industries, including finance, insurance, e-commerce, and internet banking, especially. Fraudulent activities, such as identity theft, money laundering, and credit card fraud, can result in significant financial losses for individuals and organizations. ML algorithms can analyze vast amounts of data and identify patterns that may indicate fraudulent activities. In this report, we will explore how ML is being used in fraud detection, its benefits and limitations. ML in actuality, are bunch of clever algorithms which are mostly used to find patterns in a data stream of any kind, to provide helpful information, such as the kind of fraud committed, future patterns.

1.2 Motivation:

The primary motivation for our financial fraud detection project utilising machine learning is to safeguard financial issues, protect reputations and ensure regulatory compliance. By implementing this system, we aim to enhance customer confidence, minimize operational costs and stay ahead of evolving fraud techniques. Our data-driven approach real time detection, scalability and continuous improvement, making it an effective and efficient solution for detection and preventing financial fraud, ultimately preserving trust in financial systems and institutions.

1.3 Objective:

The main objective of our financial fraud detection project using machine learning is to develop a robust and efficient system that detects and prevents fraudulent activities in financial transactions. Our goal is to protect financial assets, protect financial assets, preserve trust and reputation, ensure regulatory compliance and enhance regulatory compliance. By employing data-driven techniques, we aim to minimize operational costs, adapt to evolving fraud methods, provide real time detection, and continually improve our fraud detection capabilities, ultimately contributing to a secure and trustworthy financial environment.

1.4 Scope:

The scope of our financial fraud detection project using machine learning includes the collection and preprocessing of financial transaction data, the development of machine learning models for real time anomaly detection, integration into existing financial systems, continuous adaption to evolving fraud patterns, compliance with regulatory requirements, user training and rigorous testing to ensure the system's accuracy and efficiency. This project's primary focus is to safeguard financial assets, preserve trust, and provide a secure environment by detecting and preventing financial fraud while efficiently managing resources.

1.5 Project Outline

Chapter 1	Introduction
Chapter 2	Literature Survey
Chapter 3	System study and analysis
Chapter 4	Technologies
Chapter 5	Methodology
Chapter 6	Implementation
Chapter 7	Testing
Chapter 8	Output Screenshots
Chapter 9	Conclusion, Future Work and References

CHAPTER - 2

LITERATURE SURVEY

2.1 Literature Survey

Title: A survey of Credit Card Fraud Detection Techniques

Abstract:

In this paper, after investigating difficulties of credit card fraud detection, we seek to review the states in credit card fraud detection techniques, data sets and evaluation criteria. The advantages and disadvantages of fraud detection methods are enumerated and compared. Furtherly, a classification of mentioned techniques into two main fraud detection approaches, namely, supervised and unsupervised is presented. Again, a classification of techniques is done based on numerical and categorical data sets. Different data sets used in literature are then described and grouped into real and synthesized data and the effective and common attributes are extracted for further usage.

Title: Solving the False Positives problem in fraud prediction

Abstract:

In this paper, we present an automated feature engineering based approach to reduce false positives in fraud prediction. False positives plague the fraud prediction industry. It is estimated that only 1 in 5 declared as fraud are actually fraud and roughly 1 in every 6 customers have had a valid transaction declined in the past year. To address this problem, we use the Deep Feature Synthesis algorithm to automatically derive behavioural features based on the historical data of the card associated with a transaction. we use a random forest to learn a classifier. We tested our machine learning model on data from a large multinational bank and compared it to their existing solution.

Title: Support Vector Machines and malware detection

Abstract:

In this research, we test three advanced malware scoring techniques that have shown promise in previous research, namely, Hidden Markov Models, Simple Substitution Distance and Opcode Graph based detection. We then perform a careful robustness analysis by employing morphing strategies that cause each score to fail. We show that combining scores using a Support Vector Machine yields results that are significantly more robust than those obtained using any of the individual scores to fail. We show that combining scores using a Support Vector Machine yields results that are significantly more robust than those obtained using any of the individual scores.

Title: A Survey on Fraud Detection

Abstract:

This paper applies multiple ML techniques based on Logistic regression and support vector machine to the problem of payments fraud detection using a labelled dataset containing payment transactions and shows that these approaches are able to detect fraud transactions with high accuracy and reasonably low number of false positives.

CHAPTER - 3

SYSTEM STUDY AND ANALYSIS

3.1 Problem Statement:

Financial fraud is a persistent threat, causing significant financial losses and eroding trust. The challenge is to create an effective machine learning based system for real time detection and prevention of fraudulent transactions. This system should minimise financial losses, ensure trust, regulatory compliance and adapt to evolving fraud tactics while keeping operational costs in check. It must be scalable, easily integrated and continuously improved through data driven insight.

3.2 Existing System

Financial fraud detection system that uses machine learning have made significant advancements but they still face several limitations and challenges. These limitations include:

Linear Decision Boundaries:

Some Machine Learning models assumes a linear relationship between input features and the log-odds of the output. This means it may not perform well when the relationship between features and the target is highly non-linear.

Overfitting:

Overfitting to training data is a problem for some complex machine learning models, which makes them less generic to new data. Overfitting may arise in case of high-dimensional feature spaces, which are common in modern datasets.

Imbalanced Data:

When dealing with imbalanced datasets, where one class has significantly fewer samples than the other, some models may struggle to correctly classify the minority class. Additional techniques may often needed to address this issue.

Scalability:

Effective system is needed to handle large number of transactions. High-frequency trading and the sheer volume of financial transactions require scalable solutions that can handle large datasets and process transactions in real-time.

Concept Drift:

Fraudsters constantly adapt to new strategies, which can lead to concept drift. Models may become less effective over time as they struggle to adapt to evolving fraudulent behavior.

False Positives and Negatives:

Striking a balance between minimizing false positives (legitimate transactions flagged as fraud) and false negatives (fraudulent transactions missed) is difficult. Overly cautious systems may inconvenience genuine users, while overly lenient ones may miss fraud.

Unlabeled Data:

Anomaly detection-based systems often require labeled data for training. Creating such labeled datasets can be labor-intensive, and obtaining sufficient labeled examples of fraud can be problematic.

3.4 Proposed System:

Preparing dataset, training the model, performance evaluation and result analysis are the few activities in our proposed model. Let's go deeper into the procedure.

Step1: Obtaining and Preparing datasets:

Prepare the dataset in the appropriate directories with separate subdirectories for each class. The dataset should be split into training and testing sets so that the performance of the model can be efficiently analyzed.

Step2: Preprocessing:

Preprocessing is the critical step in the financial fraud detection using Machine learning to get the cleaned data for effective model training and to enhance the predictive capacity.

Step3: Feature Engineering:

Feature engineering is a creative process in the field of machine learning and data science. It involves selecting, transforming and creating relevant input variables (features) for a predictive model. Effective feature engineering can significantly impact the performance of a machine learning model.

Step4: The model Architecture's Personalization:

Credit card transactions are highly unbalanced where legitimate transactions may dominate over fraudulent transactions. To build the effective model handling unbalanced data is necessary. It can be achieved by applying oversampling techniques(SMOTE).

Step5: Training the model:

The supervised Machine Learning algorithm Random Forest is used to effectively and efficiently identify fraudulent transactions. Training a model on credit card fraud detection requires pre-processed data without any errors and fine tuning to make the model user-friendly.

Step6: Evaluation of Performance and results:

The model performance is evaluated based on metric scores like accuracy, precision, recall and f1-score. One can fine tune the model based on confusion matrix which depicts the false positives and false negatives if any.

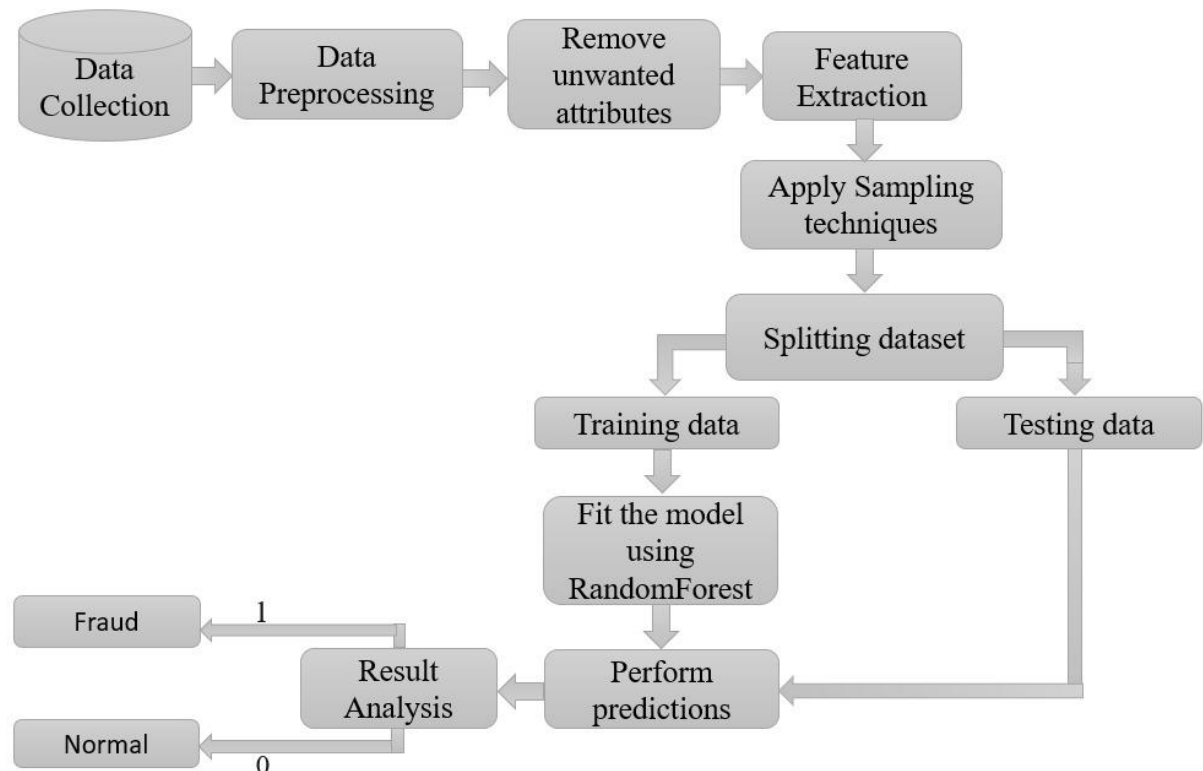


Fig.1 Proposed System

3.4.1 Random Forest

Random forest is a popular ensemble learning technique in machine learning. It's a versatile and powerful algorithm known for its ability to handle various types of tasks, including classification and regression.

1. Ensemble Learning

Random Forest is an ensemble method that combines the predictions of multiple decision trees to make more accurate and robust predictions. It's like a "forest" of decision trees working together.

2. Decision Trees

Each tree in a Random Forest is a decision tree. Decision trees are simple, hierarchical structures that make decisions by splitting the data into subsets based on the features. Random Forest uses a collection of these trees.

3. Randomness

The "random" part of Random Forest comes from two sources of randomness:

- **Random Sampling:** When building each tree, it selects a random subset of the training data (with replacement). This process is called bagging or bootstrap aggregating.
- **Random Subset of Features:** At each node in a tree, it considers only a random subset of features for splitting. This helps prevent overfitting and promotes diversity among the trees.

4. Voting or Averaging

For classification tasks, Random Forest combines the predictions of individual trees through a majority vote. In regression tasks, it averages the predictions to produce a final output.

5. Robustness

Random Forest is robust to overfitting, as the combination of many trees reduces the risk of making decisions based on noise in the data.

6. Feature Importance

It can measure the importance of each feature in the prediction, helping you understand which features are most influential.

7. Scalability

Random Forest can handle large datasets with high dimensionality and can be parallelized for faster training.

8. Wide Applicability

It's widely used in a variety of applications, including fraud detection, recommendation systems, image classification, and more.

3.5 Advantages of Proposed System:

1.High Predictive Accuracy

Random Forest often provides excellent predictive accuracy due to its ensemble of decision trees. It mitigates overfitting, resulting in more accurate and reliable predictions.

2. Robust to Overfitting

The ensemble nature of Random Forest reduces Overfitting, making it less sensitive to noise and outliers in the data, enhancing model robustness.

3. Feature Importance

It can rank the importance of input features, helping in feature selection and understanding the variables that most influence the model's predictions.

4. Handle Missing Data

Random forest can effectively handle missing values in the dataset without requiring extensive preprocessing, ensuring accurate predictions even with incomplete data.

5. Non linearity and Interaction

It can capture complex non-linear relationships and interactions between features making it suitable for a wide range of complex problems.

6. Versatility

Random Forest is versatile, applicable to both classification and regression tasks. It can work with diverse data types and distributions, making it a valuable choice for various machine learning problems.

3.6 Functional Requirements

Data Collection and Integration

The system should be able to gather data from various sources such as transactions, user behaviour, and external databases. It should be able to integrate and preprocess this data for analysis.

Feature Engineering

The system should automatically extract relevant features from the collected data to be used as input for the machine learning models.

Machine Learning Models

Implement various machine learning algorithms (e.g., supervised, unsupervised, anomaly detection) to identify patterns and anomalies in the financial data. Train and update the models periodically to adapt to evolving fraud patterns.

Real-time Monitoring

The system should continuously monitor incoming transactions and user behaviour in real-time to detect potential fraud.

Anomaly Detection

Identify unusual and suspicious patterns that could indicate fraud, such as unexpected transaction amounts or locations.

User Profiles and Behaviour Analysis

Develop user profiles based on historical data to better understand typical user behaviour and identify deviations.

3.7 Non- Functional Requirements:

Accuracy and Precision

The system should achieve a high level of accuracy and precision in identifying fraudulent transactions to minimize false positives and negatives.

Scalability

The system should be able to handle a large volume of transactions and data as the user base grows.

Real-time Processing

The system should process transactions in real-time to provide timely fraud detection and prevention.

Security and Privacy

Ensure that the sensitive financial data and user information are securely stored and processed to adhere to data protection regulations.

Reliability and Availability

The system should be reliable and available around the clock to prevent any disruptions in fraud detection services

3.8 Hardware Requirements:

Processor: Intel i3

RAM: 4GB

Memory: 512GB

3.9 Software Requirements:

OS: Windows 7

Programming languages: Python

Libraries:

Pandas, Scikit-learn, Seaborn, Matplotlib.pyplot, joblib, imbalanced-learn

IDE: Visual Studio Code

CHAPTER-4

SYSTEM DESIGN

4.1 System Architecture :

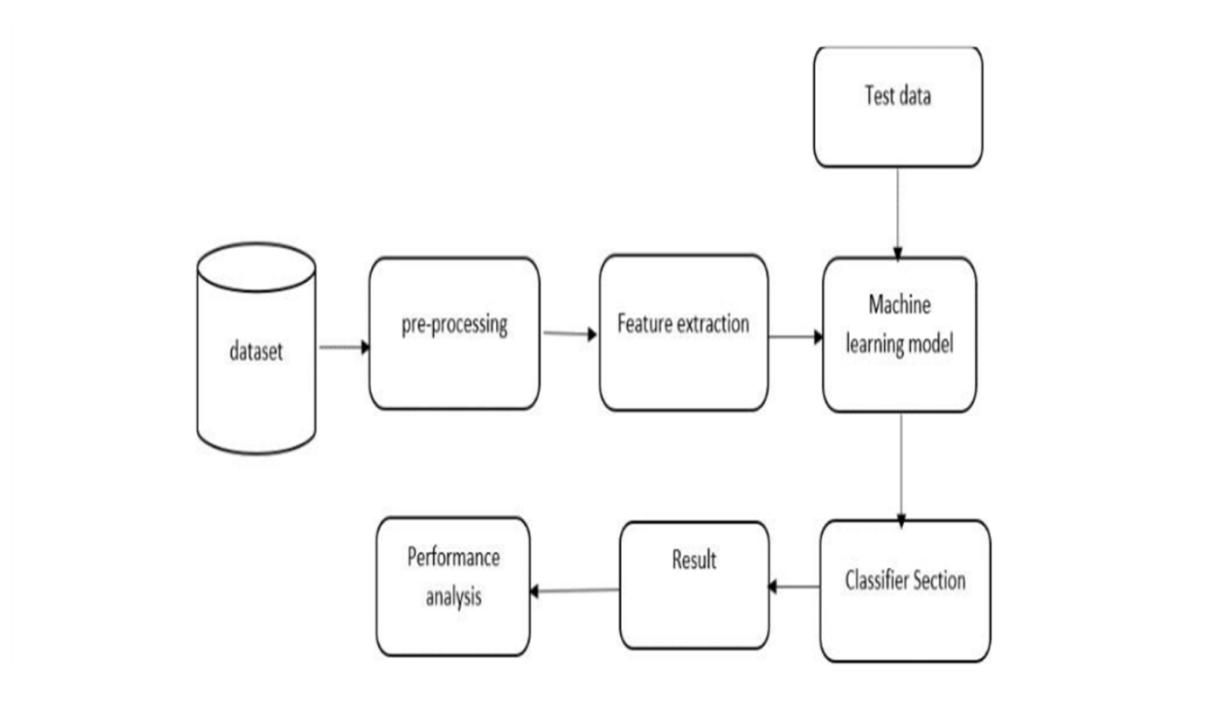


Fig.2 System Architecture

Data Collection:

Data from various sources, such as transaction records, user profiles, and historical data, is collected. This data could be both structured (like transaction amounts, dates, and locations) and unstructured (like text data from transaction descriptions).

Data Preprocessing:

Raw data is cleaned, transformed, and standardized to ensure consistency and quality. This includes handling missing values, outlier detection, and data normalization.

Feature Engineering:

The system should automatically extract relevant features from the collected data to be used as input for the machine learning models.

Machine Learning Models:

Implement various machine learning algorithms (e.g., supervised, unsupervised, anomaly detection) to identify patterns and anomalies in the financial data. Train and update the models periodically to adapt to evolving fraud patterns.

Real-time Monitoring:

The system should continuously monitor incoming transactions and user behaviour in real-time to detect potential fraud.

Anomaly Detection:

Identify unusual and suspicious patterns that could indicate fraud, such as unexpected transaction amounts or locations.

Classifier Section:

It classifies whether the transaction is fraud or non fraud.

User Profiles and Behaviour Analysis:

Develop user profiles based on historical data to better understand typical user behaviour and identify deviations.

CHAPTER- 5

TECHNOLOGIES

5.1 Python

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.

5.2 About 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 a computer programming language often used to build websites and software, automate tasks, and conduct data analysis.
- Python is a dynamic, interpreted (bytecode-compiled) language. There are no type declarations of variables, parameters, functions, or methods in source code.
- This makes the code short and flexible, and you lose the compile-time type checking of the source code.
- Python source files use the ".py" extension and are called "modules."

➤ Python was designed for readability, and has some similarities to the English language with influence from mathematics. Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses. Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

5.2 Libraries of Python

Pandas:

- i. Pandas is a data manipulation and analysis library for Python.
- ii. It provides data structures like Data Frames, which are efficient for handling structured data.
- iii. Pandas is widely used for tasks such as data cleaning, exploration, and transformation.

Scikit-learn:

- i. Scikit-learn is a machine learning library for Python.
- ii. It provides simple and efficient tools for data analysis and modelling, including various algorithms for classification, regression, clustering, and more.
- iii. Scikit-learn is designed to work seamlessly with other scientific computing libraries like NumPy and SciPy.

Matplotlib:

- i. Matplotlib is a 2D plotting library for Python.
- ii. It produces high-quality static, animated, and interactive visualizations in Python.
- iii. Matplotlib is widely used for creating plots and charts in various fields, including scientific research, data analysis, and machine learning.

Seaborn:

- i. Seaborn is a statistical data visualization library based on Matplotlib.
- ii. It provides a high-level interface for drawing attractive and informative statistical Graphics.
- iii. Seaborn simplifies the process of creating common statistical plots and enhances visual appeal of plots.

Joblib:

- i. Joblib is a library for lightweight pipelining in Python.
- ii. It provides tools for parallel computing and efficient caching of expensive function calls, particularly useful for machine learning tasks.
- iii. Joblib is commonly used in conjunction with scikit-learn for parallelizing computations.

Imbalanced-learn:

- i. Imbalanced-learn is a library for handling imbalanced datasets in machine learning.
- ii. It provides various techniques for resampling, such as over-sampling, under-sampling, and ensemble methods, to address issues when classes are not represented equally in the training data.

5.3Dependencies

Visual Studio Code:

We used VS Code as an interface here, Visual Studio Code (VSCode), developed by Microsoft, is a lightweight and cross-platform source code editor. Its appeal lies in its simplicity, speed, and rich feature set. VSCode supports various operating systems, fostering a consistent development experience. Highly extensible, the editor boasts a vast array of extensions from the Visual Studio Code Marketplace. Intelligent code editing features, such as smart code completion and syntax highlighting, enhance productivity. With an integrated terminal and debugger, developers can seamlessly execute commands and debug code within the editor. Git integration allows for version control directly within VSCode. The Live Share feature facilitates real-time collaboration, enabling pair programming and collaborative debugging. With support for a wide range of programming languages and a vibrant community contributing to its development, Visual Studio Code stands out as a versatile and community-driven code editor.

CHAPTER -6

IMPLEMENTATION

6.1 Implementation:

1.Install following libraries using pip

- pip install numpy
- pip install pandas
- pip install matplotlib.pyplot
- pip install seaborn
- pip install scikit-learn
- pip install joblib
- pip install imbalanced-learn

2.Make necessary imports

```
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot
import seaborn
import pandas
import numpy
from sklearn.model_selection import train_test_split
from imblearn.over_sampling import SMOTE
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1score
```

3.Now let's read our dataset into a Data Frame

```
data=pd.read_csv('creditcard.csv')
```

It is an imbalanced dataset, where the normal transactions are highly dominated over fraudulent transactions.

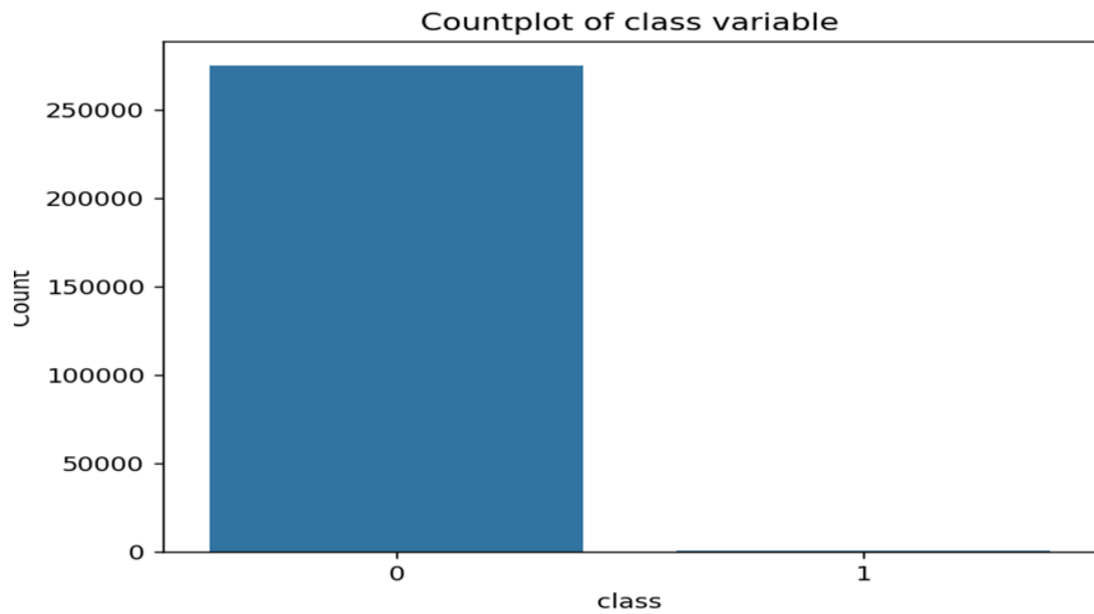


Fig.3 Count plot of class Variable

4.Perform preprocessing of the missing values and apply oversampling techniques to handle imbalanced data.

```
from imblearn.over_sampling import SMOTE
X_res,y_res=SMOTE().fit_resample(X,y)
Where X=data.drop('Class',axis=1)
y=data['Class']
```

5.Split the dataset into training and testing datasets.

```
X_train,X_test,y_train,y_test=train_test_split(X_res,y_res,test_size=0.20,random_state=42)
```

6.Now train and test the data using Random Forest Classifier algorithm

```
model=RandomForestClassifier()
model.fit(X_res, y_res)
```

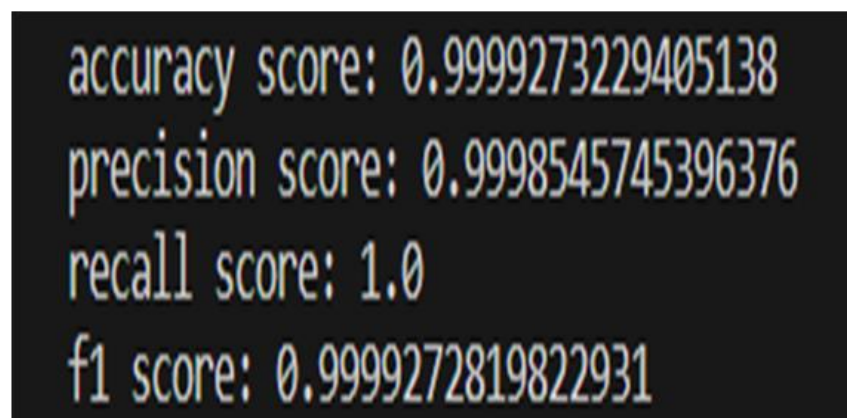
7.Saving the model with .pkl extension using joblib library

```
joblib.dump(model, 'trained_model.pkl')
```

8.In a new script file, load the trained model and make the predictions using loaded model.

```
loaded_model = joblib.load('trained_model.pkl')  
predictions = loaded_model.predict([transaction instance])
```

It gives the metrics results as follows:



```
accuracy score: 0.9999273229405138  
precision score: 0.9998545745396376  
recall score: 1.0  
f1 score: 0.9999272819822931
```

Fig.4 Metric Results

CHAPTER - 7

TESTING

7.1 Purpose of 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, subassemblies, 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 requirements.

7.2 Types of Testing:

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.

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 satisfactory, 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.

Functional testing:

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 centered on the following items:

Valid Input - identified classes of valid input must be accepted. Invalid Input

Functions - identified functions must be exercised. Output - identified classes of application outputs must be exercised.

Systems/Procedures - 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 identifying 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.

System Testing:

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 description and flows, emphasizing pre-driven process links and integration points.

White Box Testing:

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

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.

Unit Testing:

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.

7.3 Test Cases:

When testing a credit card fraud detection model, particularly one built using a Random Forest algorithm, it's important to create diverse test cases to ensure the model's effectiveness in different scenarios. Here are some test cases to consider:

1. Normal Transaction:

- Generate a test case with a typical, legitimate credit card transaction. Ensure that the features align with a standard, non-fraudulent pattern.

2. Common Fraud Patterns:

- Create test cases that mimic common fraud patterns, such as transactions from multiple countries in a short time, transactions exceeding a certain amount, or multiple small transactions in rapid succession.

It's crucial to have a mix of positive and negative cases, representing both normal and fraudulent scenarios, to thoroughly evaluate the model's performance and generalization capabilities.

Test Case 1:

Input:

```
[[-1.3598071336738, -0.0727811733098497, 2.53634673796914, 1.37815522427443, -0.338320769942518, 0.462387777762292, 0.239598554061257, 0.0986979012610507, 0.363786969611213, 0.0907941719789316, -0.551599533260813, -0.617800855762348, -0.991389847235408, -0.311169353699879, 1.46817697209427, -0.470400525259478, 0.207971241929242, 0.0257905801985591, 0.403992960255733, 0.251412098239705, -0.018306777944153, 0.277837575558899, -0.110473910188767, 0.0669280749146731, 0.128539358273528, -0.189114843888824, 0.133558376740387, -0.0210530534538215, 149.62]
```

Output:

Normal Transaction

Test Case 2:

Input:

```
[[-2.3122265423263, 1.95199201064158, -1.60985073229769, 3.9979055875468, -0.522187864667764, -1.42654531920595, -2.53738730624579, 1.39165724829804, -2.77008927719433, -2.77227214465915, 3.20203320709635, -2.89990738849473, -0.595221881324605, -4.28925378244217, 0.389724120274487, -1.14074717980657, -2.83005567450437, 0.0168224681808257, 0.416955705037907, 0.126910559061474, 0.517232370861764, 0.03593686052974, 0.46521076182388, 0.320198198514526, 0.0445191674731724, 0.177839798284401, 0.261145002567677, -0.143275874698919]
```

Output:

Fraudulent Transaction

CHAPTER-8

OUTPUT SCREENSHOTS

Let's give inputs to find whether a transaction is fraud or not.

```
pred=model.predict([[-1.3598071336738, -0.0727811733098497, 2.53634673796914, 1.37815522427443, -0.338320769942518, 0.46238777762292, 0.239598554061257, 0.0986979012610507, 0.363786969611213, 0.0907941719789316, -0.551599533260813, -0.617800855762348, -0.991389847235408, -0.311169353699879, 1.46817697209427, -0.470400525259478, 0.207971241929242, 0.0257905801985591, 0.403992960255733, 0.251412098239705, -0.018306777944153, 0.277837575558899, -0.110473910188767, 0.0669280749146731, 0.128539358273528, -0.189114843888824, 0.133558376740387, -0.0210530534538215, 149.62]])
```

OUTPUT:

Normal Transaction

```
pred=model.predict([[-2.3122265423263, 1.95199201064158, -1.60985073229769, 3.9979055875468, -0.522187864667764, -1.42654531920595, -2.53738730624579, 1.39165724829804, -2.77008927719433, -2.77227214465915, 3.20203320709635, -2.89990738849473, -0.595221881324605, -4.28925378244217, 0.389724120274487, -1.14074717980657, -2.83005567450437, -0.0168224681808257, 0.416955705037907, 0.126910559061474, 0.517232370861764, -0.0350493686052974, -0.465211076182388, 0.320198198514526, 0.0445191674731724, 0.177839798284401, 0.261145002567677, -0.143275874698919, 0]])
```

OUTPUT:

Fraudulent Transaction

OUTPUT ANALYSIS:

- This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.
- Yet, this Random Forest model predicts the output with an accuracy score of 99.9%.
- The feature 'Class' is the target variable which takes 1 in case of fraud and 0 otherwise.
- It exhibited high precision, recall, and F1-score, which are critical for minimizing false positives and false negatives in fraud detection.

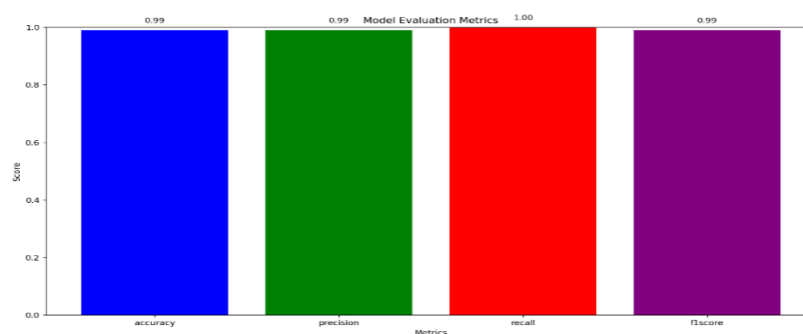


Fig.5 Output Analysis

Confusion Matrix:

A confusion matrix is a fundamental visualization to measure the capability of a classification model. It shows the true positive, true negative, false positive and false negative predictions, allowing you to see the model's accuracy, precision, recall and F1-score.

```
[[55064  0]]
```

```
[9  55003]
```

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	55073
1	1.00	1.00	1.00	55003
accuracy			1.00	110076
macro avg	1.00	1.00	1.00	110076
weighted avg	1.00	1.00	1.00	110076

Fig.6 Classification Report

ROC (Receiver Operating Characteristic) Curve:

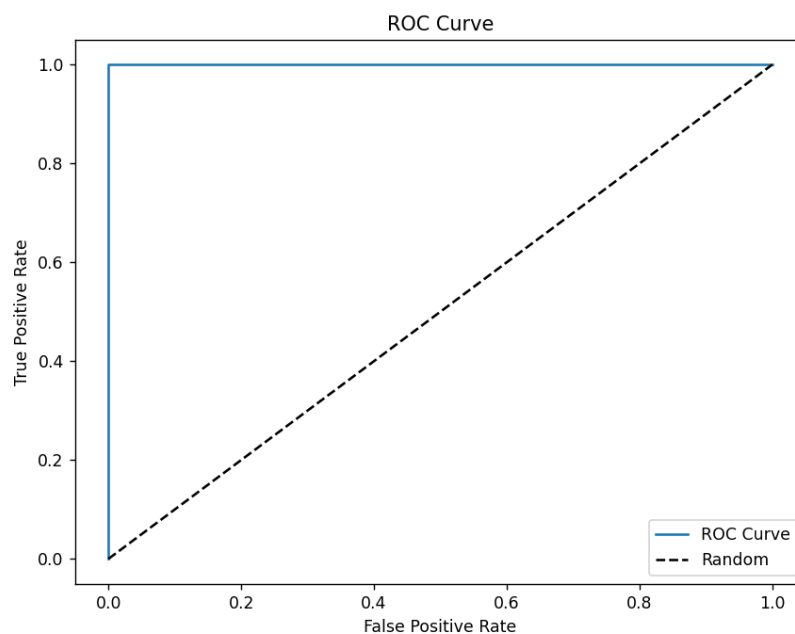


Fig.7 ROC Curve

CHAPTER-9

CONCLUSION AND FUTURE WORK

9.1 Conclusion:

Finally, Our financial fraud detection project employing a Random Forest model offers a powerful and effective approach to mitigating financial fraud risks with a remarkable accuracy off 99.9%. The ability of the Random Forest model to precisely identify financial fraud has been demonstrated. Its ensemble learning method reduces false positives and false negatives by combining multiple decision trees to produce reliable results. Despite being well-known for their ability to predict outcomes, Random Forest models also provide some interpretability, which aids fraud investigators in understanding why particular transactions are reported as possibly fraudulent. Monitoring constantly emerging frauds is a continuous process in real time.

9.2 Future Enhancement:

Future enhancements for a financial fraud detection project using machine learning will likely involve more advanced technologies and strategies. This includes using deep learning models for better pattern recognition, real time analysis to speed up detection and ensemble methods to improve accuracy. It's important to ensure the system can explain its decisions and adapt to new fraud tactics. Leveraging behavioural biometrics and external data sources can further strengthen the system. Additionally, making it more user-friendly, capable of cross channel and cross organization analysis and forecasting global collaboration will enhance its overall effectiveness in fighting financial frauds. These future improvements reflect the need for a more sophisticated and adaptable system to address evolving fraud challenges while maintaining compliance and data security. Collaborating with cybersecurity teams to address the convergence of cyber threats and financial fraud.

9.3 References:

- 1.Samaneh Sorournejad, Zojah and Atani,"A survey of Credit Card Fraud Detection Techniques",2016.
- 2.Wedge , Canter and Rubio , "Solving the False Positives problem in fraud prediction",2017.
- 3.T.Singh , F.Di Troia , C.Visaggio , "Support Vector Machines and malware detection",2015.
- 4.aditya Oza,"A Survey on Fraud Detection",2019.
- 5."Credit Card Fraud Detection: A Realistic Modelling and a Novel Learning Strategy" published by IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, VOL.29,NO.8,AUGUST 2018.
- 6.XuanS, LiuG, LiZ, ZhengL, WangS, JiangC. Random Forest credit card fraud detection. In:2018 IEEE 15th international conference on networking, sensing and control (ICNSC). IEEE; 2018.
- 7."Detection of Financial Statement Fraud: A Review of the Literature" by Elliott, R. K., & Willingham, J. J.