**DESIGN PROJECT - II REPORT**

On

**FRAUD DETECTION IN FINANCIAL TRANSACTIONS**

Submitted in Partial Fulfillment of Award of

**BACHELOR OF TECHNOLOGY**

**In**

**Computer Science and Engineering**

By

[AVATI JAYANTH KUMAR, 2022BCSE07AED032,

MAYAKUNTLA SAI SATHWIK,2022BCSE07AED033,

SHAGAM SRINIVASAREDDY,2022BCSE07AED034,

DONADULA MOUNISH KUMAR,2022BCSE07AED035,

CHOUGULE SAISH SAMIR,2022BCSE07AED040]

Under the Supervision of

**Dr Sridhar D, Professor, Computer Science and Engineering**



**ALLIANCE COLLEGE OF ENGINEERING AND DESIGN**

**ALLIANCE UNIVERSITY**

**BENGALURU**

NOVEMBER & 2024



**Department of Computer Science and Engineering**

**ALLIANCE COLLEGE OF ENGINEERING AND DESIGN**

**CERTIFICATE**

This is to certify that the Design project – II work entitled “FRAUD DETECTION IN FINANCIALTRANSACTIONS”submitted by AVATIJAYANTH KUMAR,[2022BCSE07AED032],MAYAKUNTLASAISATHWIK,[2022BCSE07AED033],SHAGAMSRINIVASAREDDY[2022BCSE07AED034],DONADULAMOUNISHKUMAR[2022BCSE07AED035],CHOUGULESAISHSAMIR,[2022BCSE07AED040]. In partial fulfillment for the award of the degree of Bachelor of Technology (Computer Science and Engineering) of Alliance University, is a bonafide work accomplished under our supervision and guidance during the academic year 2024-2025. This thesis report embodies the results of original work and studies conducted by students and the contents do not form the basis for the award of any other degree to the candidate or anybody else.

**Dr Sridhar D Dr.A.Ezil Sam Leni**

(Supervisor) (Head of Department)

**External Examiners**

1. **Name: Signature**
2. **Name: Signature**



**Computer Science and Engineering**

**ALLIANCE COLLEGE OF ENGINEERING AND DESIGN**

**DECLARATION**

I/We hereby declare that the Design project-II entitled **Fraud Detection In Financial Transactions** submitted by me/us in the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology **(Artificial Intelligence & Machine Learing)** of Alliance University, is a record of my/our work carried under the supervision and guidance of **Dr Sridhar D**.

We confirm that this report truly represents the work undertaken as a part of our project work. This work is not a replication of work done previously by any other person. We also confirm that the contents of the report and the views contained therein have been discussed and deliberated with the faculty guide.

|  |  |  |
| --- | --- | --- |
| **Name of the Student** | **University Registration Number** | **Signature** |
| AVATI JAYANTH KUMAR | **2022BCSE07AED032** |  |
| MAYAKUNTLA SAI SATHWIK | **2022BCSE07AED033** |  |
| SHAGAM SRINIVASAREDDY | **2022BCSE07AED034** |  |
| DONADULA MOUNISH KUMAR | **2022BCSE07AED035** |  |
| CHOUGULE SAISH SAMIR | **2022BCSE07AED040** |  |

**ACKNOWLEDGEMENT**

The satisfaction that accompanies the successful completion of the task would be put incomplete without

the mention of the people who made it possible, whose constant guidance and encouragement crown all

the efforts with success.

We avail this opportunity to express our deep sense of gratitude and hearty thanks to the Management of Alliance University, for providing world class infrastructure, congenial atmosphere and encouragement.

We would like to thank **Dr. Reeba Korah**, Dean, ACED and **Dr. A.Ezil Sam Leni**, Head of the department of CSE/IT, for and their encouragement and cooperation at various levels of the project.

We are very thankful to our Guide **Dr.Sridhar D** and Project Coordinators **Dr. M. Selvam and Dr. Lopa Mandal**, Department of Computer Science and Engineering & Information Technology for their sustained inspiring guidance and cooperation throughout the process of this project. Their wise counsel and valuable suggestions are invaluable.

We express our deep sense of gratitude and thanks to the teaching and non-teaching staff at our department who stood with us during the project and helped us to make it a successful venture.

We place highest regards to my parents, my friends and well-wishers who helped a lot in making the report of this project.

AVATI JAYANTHKUMAR,

MAYAKUNTLA SAI SATHWIK,

SHAGAM SRINIVASA REDDY,

DONADULA MOUNISH KUMAR,

CHOUGULE SAISH SAMIR.

**PREFACE**

From the emergence of E-banking, E-commerce and mobile payments, the speed and the amount in which financial transactions occur in today’s market has never been seen before. But this rise has also matched with the growth of fraudulent activities that pose serious threats to individuals, businesses and financial institutions. Fraudulent transactions are of grave concern for organizations and businesses as they lead to huge financial losses, harm data as well as institution’s reputation hence the importance of fraud detection in the market.Fraud detection in financial transaction processes or systems involves flagging any activity that raises any concern or is out of the transaction norms such as the amount of the transaction, time or even the location. The use of advanced analytics, machine learning and AI nowadays enables the company or organization to be able to reduce or completely prevent the escalated situation of fraud. The contemporary models of dealing with fraud identification rest on various approaches such as statistical methods, supervised and unsupervised learning, anomaly detection and network analysis. These approaches facilitate continuous surveillance and decision making which greatly improves the efficiency and effectiveness of fraud detection.

This preface explains different styles of fraud detection mechanisms, outlining the traditional rules based systems and the newer machine learning systems. Readers through this survey will be informed of the difficulties and the advances achieved in combating fraud, the significance of the data, as well as the moral challenges of applying AI to financial security. In such regard, the purpose of this work is to educate the audience on the tools and approaches that influence the growth of fraud management policies within the financial industry with a hope of making it safer.

**ABSTRACT**

Recognizing extortion in monetary exchanges is pivotal for defending against progressively advanced false exercises in today's advanced age. As computerized exchanges have become more predominant, budgetary teach experience more prominent challenges in distinguishing and anticipating extortion. This requires the appropriation of progressed innovations and imaginative techniques to maintain the keenness of exchanges and ensure both businesses and shoppers from money related hurt and reputational damage.

The changing scene of the extortion discovery in monetary exchanges, investigating the most recent patterns, advances, and techniques utilized by budgetary educate and administrative bodies. Through the utilization of machine learning, fake insights, and information analytics and organizations are able analyze huge volumes of value-based information in real-time and empowering the location of the unpretentious peculiarities and deviations from typical behavior. Furthermore, behavioral analytics plays a imperative part in extortion location by building up commonplace action designs for each account holder and distinguishing deviations that may show potential false activities.

Collaboration and data sharing are the two imperative components in combating extortion. Money related educate, installment processors, law authorization offices, and administrative bodies must collaborate to share experiences, insights, and best hones for the extortion discovery and anticipation. By cultivating collaboration and leveraging collective skills, partners are able to move forward their capacity to identify and relieve extortion exercises, subsequently keeping up great belief in the budgetary framework.

**KEYWORDS:**   
1.Blackmail recognition  
2.Payments  
3.Extortion detection  
4.Beta-testing of designs  
5.Machine learning   
6.Artificial Intelligence  
7.Information Retrieval  
8.Behavior analysis  
9.Continuous surveillance.

**TABLE OF CONTENTS**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SL.NO | TOPICS | | | | | | | | Page.No |
| 1. | **CERTIFICATE** | | | | | | | |  |
| 2 | **DECLARATION** | | | | | | | |  |
| 3 | **ACKNOWLEDGMENT** | | | | | | | |  |
| 4 | **PREFACE** | | | | | | | |  |
| 5 | **ABSTRACT** | | | | | | | |  |
| 6 | **1.INTRODUCTION** | | | | | | | |  |
| 7 | **2.** **LITERATURE SURVEY** | | | | | | | |  |
| 2.1 | | | | | Literature review | | |  |
| 2.2 | | | | | Limitations of the existing system | | |  |
| 2.3 | | | | | Scope of the project | | |  |
| 8 | **3.SYSTEM DESIGN** | | | | | | | |  |
| 3.1 | | | | | Problem Definition | | |  |
| 3.2 | | | | | System Architecture | | |  |
| 3,3 | | | | | Requirement Specifications | | |  |
|  | | | | | 3.3.1 | | Hardware Requirements |  |
| 3.3.2 | | Software Requirements |  |
| 9 | **4.SYSTEM IMPLEMENTATION** | | | | | | | |  |
| 4.1 | | | Overview of the modules | | | | |  |
| 4.2 | | | Description of the modules | | | | |  |
|  | | | 4.2.1 | | | Module 1 description | |  |
| 4.2.2 | | | Module 2 description | |  |
| 4.2.3 | | | Module 3 description | |  |
| 10 | **5. RESULT AND CONCLUSION** | | | | | | | |  |
| 5.1 | | Description | | | | | |  |
| 5.2 | | Graphs | | | | | |  |
| 11 | **6.TESTING** | | | | | | | |  |
| 6.1 | | | | Description | | | |  |
| 6.2 | | | | Test cases | | | |  |
| 12 | **7.CONCLUSION AND FUTURE ENHANCEMENTS** | | | | | | | |  |
| 7.1 | Conclusion | | | | | | |  |
| 7.2 | Limitations of the project | | | | | | |  |
| 7.3 | Advantages of the project | | | | | | |  |
| 7.4 | Applications of the project | | | | | | |  |
| 7.5 | Future enhancements | | | | | | |  |

# CHAPTER 1

# INTRODUCTION

Financial services include fraud detection and Prevention, assistance forbids them or their employ from abominable actions thus enabling the safety of the entirety of the financial systems. Credit card fraud – meaning the unauthorized use of a credit card or knowing its details to make a fraudulent transactions, identity theft, money laundering, or insider trading; these activities alongside others pose threats not only to institutions, but also to individual consumers with severe economic and reputational ramifications. Statistics from the analysis show that financial fraud sweeps billions from the global economy annually and holds severe consequences on private clients as well as businesses.  
With the relative ease and convenience of transacting over the internet, criminals are not known to become complacent, but rather develop sophisticated ways of scamming people as they exploit loopholes within these financial systems. This advanced mother and scope necessitated the development of automated fraud detection systems that would employ the extraction and analysis of large volumes of data in the shortest time and real time; engaging machine learning and artificial intelligence AI and big data analytics, we can spot unusual behavior and stop such transactions.

Traditional systems that rely on rules to detect fraud lacks efficiency because even criminals have their own developments, in other words they evolve, thus new criminal activities require new methods of engagement. This is the key reason why fraud detection models are beginning to use more dynamic machine learning algorithms. Such algorithms have the capabilities to improve themselves progressively. Such models rely on previous data on transaction behavior to categorize a given transaction into either normal or abnormal, thus enhancing the accuracy of the system and reducing false alarms. It is important to note that the current circumstances are quite different; there is a greater requirement to combat fraud as rapidly as funds change hands in considerable volumes across many platforms when looking at the frameworks. In the current cyber world, advancing technology brings forth their own challenges, false information offers criminals new opportunities to exploit weaknesses in financial organizations and their institutions. From instances such as fraudulent credit cards or theft of identity and even money laundering, the entire financial industry is under different threats that require new detection techniques.

**CHAPTER 2**

# LITERATURE SURVEY

## 2.1 LITERATURE REVIEW

This literature review investigates some of the works carried out in the past few years on the use of machine learning in the detection and prevention of frauds in the financial institutions with an emphasis on the newer technologies that are more accurate and capable of dealing with complicated frauds.It has been established in some studies that a variety of fraud discovery models and algorithms grounded on machine literacy styles including deep literacy, ensemble styles, and anomaly discovery, are indeed pivotal in demonstrating these suspicious sale patterns. For case, CNN( Convolution Neural Networks) and RNN( intermittent Neural Networks) models which are distributed as deep literacy models have shown a better delicacy position in detecting complex patterns in the transactional data than the traditional styles, therefore further supporting the argument that they're superior in terms of performance and effectiveness Focusing in the reverse direction, this area of research looks into where verification could come from blockchain technology, which possesses attributes such as immutability, decentralization, and transparency that allow for the recording of transactions in a more secure and traceable manner. Examples include tamperproof records, real-time records, and peer-to-peer transactions which are secure, all of which reduce the chances of fraud and are incorporated into blockchain based systems. A combined approach which is rule-based, machine learning and anomaly detection has also shown effectiveness as it incorporates strengths of various approaches to improve on the accuracy and performance in detecting new and more malleable fraud patterns.

Furthermore, big data analytics has a great impact on fraud detection through the examination of transaction databases, fraud detection systems that are capable of predictive analytics using machine learning over a range of data sources. In the same vein, graph-based approaches tackle the challenge of fraud in a transactional context by focusing on the patterns and irregularities of transactive relations, which makes it possible to expose complex schemes of fraud and links that tie the perpetrators of such crimes. Along with that The authors have also investigated the work on the metaheuristics optimizers such as the genetic algorithms, particle swarm optimization, and simulated annealing, for the purpose of tuning the parameters of the fraud detection models, which improved their capabilities, and deployment.

**2.2 Limitations of the Existing System**

* **Scalability:** Many models struggle to handle the growing volume of financial transactions in real time, which limits efficiency as the volume of transactions increases.
* **Lack of interpretability:** Deep learning, while effective, is often “black boxes”, making it difficult for analysts to understand why specific transactions are flagged.
* **High Computational Costs:** Resource intensive models like deep learning and ensemble techniques require significant computational power, making scaling costly.
* **Integration Complexity:** Integrating blockchain and real-time streaming analytics into fraud detection is complex, limiting the full use of these technologies.

**2.3 Scope of the Project**

This supplement seeks to vitalize the authority by introducing cardholders settlement and further attempts to address these drawbacks by developing a constructive blackmail detection scheme. The extension highlights the stable identification of fictitious and flexible exchanges integration of machines learning and blockchain with real time analytics. However, there will be cost-effective optimization approaches to the need in coming up with metaheuristic techniques with a real time interface to the productivity and dispatch exercise. Exceptionally, the extension will also focus on the development of interpretable models which correspond to the timing and accuracy of retrospective budgetary decision making. Further, inclusive and tight integration of blockchain for secure exchange tracing and leakage analytics for systematic recovery of extortion location will enable convergence of both expanding technologies to increase extortion expectations in monetary environments. Huge amounts of exchange and adjusting to the increasing supply for new electronic payment systems and internet trade.

**3.SYSTEM DESIGN**

A financial transaction fraud detection system identifies unusual patterns by analyzing transaction data in real time. Using machine learning models, it detects anomalies and helps banks and financial institutions effectively prevent fraudulent activities. The main objective of this system is to detect and prevent fraudulent financial transactions. This includes real-time analysis of massive amounts of transaction data to identify anomalies and suspicious patterns within them.

**3.1 PROBLEM DEFINITION**

* The main ideal of this system is to descry and help fraudulent fiscal deals. This includes real- time analysis of massive quantities of sale data to identify anomalies and suspicious patterns within them. system should be able to:
* Real- time Discovery Identify fraudulent deals incontinently as they do.
* Accurate Discovery Minimizing the false cons and negatives.
* Scalability Handle adding volumes of data and deals.
* Rigidity This means that to learn from new fraud patterns and acclimate discovery models consequently. User-Friendliness: Provide a user-friendly interface for analysts to monitor alerts and investigate suspicious activity.

**Crucial challenges:**

* Volume and speed of data the system needs to reuse a huge quantum of data in real time, which ensures fast processing and analysis.
* Evolving Fraud Tactics Fraudsters are constantly developing new ways, making it delicate to stay ahead of new ways.
* Data quality This means that maintaining accurate, complete and high- quality data is critical to the trustability and effectiveness of fraud discovery.
* Balancing and recall delicacy the system must directly identify fraudulent deals while minimizing false cons and negatives.

**3.2 SYSTEM ARCHITECTURE**

A diagram of a diagram

Description automatically generated

The framework design is planned for real-time extortion discovery amid card exchanges. The handle starts with installment information entering the framework, which is at that point prepared by a extortion location demonstrate through Apache Flink, effective real-time information handling capabilities that empower quick investigation of approaching installment information to identify potential extortion promptly after a exchange. happens. the engineering incorporates components such as a installment interface, real-time analytics utilizing Flink, a card extortion location show, and an alarming and visualization framework. These components work together to rapidly distinguish false exchanges and inform cardholders.The framework design is planned for real-time extortion location amid card exchanges. The handle starts with installment information entering the framework, which is at that point prepared by a extortion discovery demonstrate through Apache Flink, capable real-time information preparing capabilities that empower quick investigation of approaching installment information to distinguish potential extortion instantly after a exchange happens. the design incorporates factors analogous as a installment interface, real- time analytics exercising Flink, a card trace thievery discovery demonstrate, and an intimidating and visualization frame. These factors work together to swiftly distinguish false exchanges and inform cardholders.

**3.3 Requirement Specifications**

Requirement specifications for a fraud detection system include the ability to analyze large volumes of transaction data in real-time and accurately identify potential fraudulent activities. The system should ensure high detection accuracy while minimizing false positives to maintain user trust and operational efficiency.

**3.3.1 Hardware Requirements**

* Server Hardware : High-performance servers for running real – time data processing.
* Storage : Sufficient storage for historical transaction data to enable accurate model training and storage of results.
* Network : Reliable and fast network infrastructure to support real-time data flow.
* Notification Infrastructure: Systems to send notifications, such as a secure SMS gateway or email server.

**3.3.2 Software Requirements**

* Operating System: Linux or Windows for server compatibility.
* Programming Language: Python, Java, or Scala (for model development and real-time analysis in Flink).
* Apache Flink: For real-time data streaming and analysis.
* Machine Learning Frameworks: Libraries like TensorFlow, PyTorch, or Scikit-Learn for model development.
* Database: A SQL or NoSQL database for storing transaction data and model predictions.
* Dashboard: A dashboard solution like Grafana or Power BI for visualizing real-time data.
* Notification System: Integration with SMS/email API to alert cardholders of fraudulent transactions.

**4.System Implementation**

This system is designed to detect fraud in card transactions in real time. It processes payment data, assesses the legitimacy of each transaction and immediately informs the cardholder if fraud is detected. The system also enables visualization of transaction data that can be used for both monitoring and future analysis.

4.1 Overview of the Modules

The system comprises of five center modules, each serving a particular reason in the blackmail location workflow:  
1. Installment Collection Module: Collects trade data from different sources for installments such as flexible contraptions, ATMs and online installment systems.

2. Real-time Examination With Flink Module : Forms exchange information in real-time utilizing Apache Flink, analyzing it to distinguish possibly false transactions.  
3. Card Extortion Location Show Module(Via REST API) : A machine learning-based demonstrate that classifies exchanges as authentic or false based on exchange features.  
4. Notice & Reaction Module : Cautions cardholders of any exchanges classified as fraudulent.  
5. Visualization and Capacity Module : Stores exchange information and gives a dashboard interface for visualizing exchange history and extortion analysis.

**4.2 Description of the Modules**

**4.2.1 Module-1 Description:** (Payment collection )

- Purpose : This module is responsible for collecting payment transaction data from various sources, including mobile apps, ATMs, and online banking platforms. It consolidates this data into a consistent format before feeding it into the real-time analysis system.

- Input : Incoming transaction data from different sources.

- Output : Provides formatted transaction data to the Real-time Analysis Module.

**DATA SET NAME: ('/content/Fraud transaction.csv')**

**4.2.2 Module 2 Description** (Real-time Analysis Module)

- Purpose : Using Apache Flink, this module analyzes incoming transactions in real time to determine if they need further scrutiny for fraud detection. It acts as the primary data processing engine, handling large volumes of transaction data efficiently and forwarding each transaction to the fraud detection model.

- Input : Transaction data from the Data Ingestion Module.

- Output: Passes analyzed transaction data to the Card Fraud Detection Model for classification as legitimate or fraudulent.

**4.2.3Module3Description(**FraudDetectionModel)  
 Reason : This module runs a machine learning show prepared to distinguish false exchanges. It takes the prepared exchange information and classifies each exchange as either authentic or false. The demonstration can utilize a assortment of highlights, such as exchange area, sum, and time, to make itsdecision.  
 Input:Analyzed exchange information fromtheReal-time Examination Module.  
- Output: Classification of each exchange (genuine or false), which is passed on to the Notice and Visualization Modules.

**Logistic Regression Implementation:**

import pandas as pd

import numpy as np

import time

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

# Load and preprocess the data

try:

    data = pd.read\_csv('/content/Fraud transaction.csv')

    print("TC-001: Data ingestion - Pass")

except Exception as e:

    print("TC-001: Data ingestion - Fail")

    print(f"Error: {e}")

# Dropping ID column if it exists

data = data.drop(columns=['id'], errors='ignore')

# Define features and target

X = data.drop(columns=['Class'])

y = data['Class']

# Split data into train and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Scaling features

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Initialize and train the model

log\_reg\_model = LogisticRegression(max\_iter=10000, random\_state=42)

# Test Case TC-002: Verify processing time (simulating real-time processing)

start\_time = time.time()

log\_reg\_model.fit(X\_train, y\_train)

end\_time = time.time()

processing\_time = end\_time - start\_time

if processing\_time < 1:

    print("TC-002: Real-time processing - Pass")

else:

    print("TC-002: Real-time processing - Fail")

# Predictions and model evaluation

y\_pred = log\_reg\_model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

# Test Case TC-003: Model accuracy threshold

if accuracy > 0.95:

    print("TC-003: Fraud detection model accuracy - Pass")

else:

    print("TC-003: Fraud detection model accuracy - Fail")

# Test Case TC-004: False positive rate threshold

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

tn, fp, fn, tp = conf\_matrix.ravel()  # Extract values from the confusion matrix

false\_positive\_rate = fp / (fp + tn)

if false\_positive\_rate < 0.05:

    print("TC-004: False positive rate - Pass")

else:

    print("TC-004: False positive rate - Fail")

# Print additional results for verification

print("\nModel Accuracy:", accuracy)

print("Classification Report:\n", classification\_report(y\_test, y\_pred))

print("Confusion Matrix:\n", conf\_matrix)

print("False Positive Rate:", false\_positive\_rate)

# Visualize test results

labels = list(test\_results.keys())

results = [1 if result == 'Pass' else 0 for result in test\_results.values()]

colors = ['green' if result == 1 else 'red' for result in results]

**K-Nearest Neighbor implementation**

import pandas as pd

import numpy as np

import time

from sklearn.model\_selection import train\_test\_split

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

from sklearn.preprocessing import StandardScaler

import matplotlib.pyplot as plt

import seaborn as sns

# Load and preprocess the data

try:

    data = pd.read\_csv('/content/Fraud transaction.csv')

    print("TC-001: Data ingestion - Pass")

except Exception as e:

    print("TC-001: Data ingestion - Fail")

    print(f"Error: {e}")

# Dropping ID column if it exists

data = data.drop(columns=['id'], errors='ignore')

# Define features and target

X = data.drop(columns=['Class'])

y = data['Class']

# Split data into train and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Scaling features

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Initialize and train the KNN model

knn\_model = KNeighborsClassifier(n\_neighbors=5)

# Test Case TC-002: Verify processing time (simulating real-time processing)

start\_time = time.time()

knn\_model.fit(X\_train, y\_train)

end\_time = time.time()

processing\_time = end\_time - start\_time

if processing\_time < 1:

    print("TC-002: Real-time processing - Pass")

else:

    print("TC-002: Real-time processing - Fail")

# Predictions and model evaluation

y\_pred = knn\_model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

# Test Case TC-003: Model accuracy threshold

if accuracy > 0.95:

    print("TC-003: Fraud detection model accuracy - Pass")

else:

    print("TC-003: Fraud detection model accuracy - Fail")

# Test Case TC-004: False positive rate threshold

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

tn, fp, fn, tp = conf\_matrix.ravel()

false\_positive\_rate = fp / (fp + tn)

if false\_positive\_rate < 0.05:

    print("TC-004: False positive rate - Pass")

else:

    print("TC-004: False positive rate - Fail")

# Print additional results for verification

print("\nModel Accuracy:", accuracy)

print("Classification Report:\n", classification\_report(y\_test, y\_pred))

print("Confusion Matrix:\n", conf\_matrix)

print("False Positive Rate:", false\_positive\_rate)

# Visualize test results with a bar chart

test\_results = {

    'TC-001: Data ingestion': 'Pass' if not data.empty else 'Fail',

    'TC-002: Real-time processing': 'Pass' if processing\_time < 1 else 'Fail',

    'TC-003: Model accuracy > 95%': 'Pass' if accuracy > 0.95 else 'Fail',

    'TC-004: False positive rate < 5%': 'Pass' if false\_positive\_rate < 0.05 else 'Fail'

}

labels = list(test\_results.keys())

results = [1 if result == 'Pass' else 0 for result in test\_results.values()]

colors = ['green' if result == 1 else 'red' for result in results]

plt.figure(figsize=(10, 6))

plt.barh(labels, results, color=colors)

plt.xlabel("Test Result (Pass=1, Fail=0)")

plt.title("Test Case Results for KNN Model")

for i, v in enumerate(results):

    plt.text(v - 0.03, i, "Pass" if v == 1 else "Fail", color="white", va="center", fontweight="bold")

plt.xlim(-0.5, 1.5)

plt.show()

# Visualize the confusion matrix with a heatmap

plt.figure(figsize=(8, 6))

sns.heatmap(conf\_matrix, annot=True, fmt="d", cmap="Blues", cbar=False,

            xticklabels=['Non-Fraud', 'Fraud'], yticklabels=['Non-Fraud', 'Fraud'])

plt.xlabel("Predicted Labels")

plt.ylabel("True Labels")

plt.title("Confusion Matrix for KNN Model")

plt.show()

**RandomForest Implementation**

import pandas as pd

import numpy as np

import time

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

from sklearn.preprocessing import StandardScaler

import matplotlib.pyplot as plt

import seaborn as sns

# Load and preprocess the data

try:

    data = pd.read\_csv('/content/Fraud transaction.csv')

    print("TC-001: Data ingestion - Pass")

except Exception as e:

    print("TC-001: Data ingestion - Fail")

    print(f"Error: {e}")

# Dropping ID column if it exists

data = data.drop(columns=['id'], errors='ignore')

# Define features and target

X = data.drop(columns=['Class'])

y = data['Class']

# Split data into train and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Scaling features

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Initialize and train the Random Forest model

rf\_model = RandomForestClassifier(n\_estimators=100, random\_state=42)

# Test Case TC-002: Verify processing time (simulating real-time processing)

start\_time = time.time()

rf\_model.fit(X\_train, y\_train)

end\_time = time.time()

processing\_time = end\_time - start\_time

if processing\_time < 1:

    print("TC-002: Real-time processing - Pass")

else:

    print("TC-002: Real-time processing - Fail")

# Predictions and model evaluation

y\_pred = rf\_model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

# Test Case TC-003: Model accuracy threshold

if accuracy > 0.95:

    print("TC-003: Fraud detection model accuracy - Pass")

else:

    print("TC-003: Fraud detection model accuracy - Fail")

# Test Case TC-004: False positive rate threshold

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

tn, fp, fn, tp = conf\_matrix.ravel()

false\_positive\_rate = fp / (fp + tn)

if false\_positive\_rate < 0.05:

    print("TC-004: False positive rate - Pass")

else:

    print("TC-004: False positive rate - Fail")

# Print additional results for verification

print("\nModel Accuracy:", accuracy)

print("Classification Report:\n", classification\_report(y\_test, y\_pred))

print("Confusion Matrix:\n", conf\_matrix)

print("False Positive Rate:", false\_positive\_rate)

# Visualize test results with a bar chart

test\_results = {

    'TC-001: Data ingestion': 'Pass' if not data.empty else 'Fail',

    'TC-002: Real-time processing': 'Pass' if processing\_time < 1 else 'Fail',

    'TC-003: Model accuracy > 95%': 'Pass' if accuracy > 0.95 else 'Fail',

    'TC-004: False positive rate < 5%': 'Pass' if false\_positive\_rate < 0.05 else 'Fail'

}

labels = list(test\_results.keys())

results = [1 if result == 'Pass' else 0 for result in test\_results.values()]

colors = ['green' if result == 1 else 'red' for result in results]

plt.figure(figsize=(10, 6))

plt.barh(labels, results, color=colors)

plt.xlabel("Test Result (Pass=1, Fail=0)")

plt.title("Test Case Results for Random Forest Model")

for i, v in enumerate(results):

    plt.text(v - 0.03, i, "Pass" if v == 1 else "Fail", color="white", va="center", fontweight="bold")

plt.xlim(-0.5, 1.5)

plt.show()

# Visualize the confusion matrix with a heatmap

plt.figure(figsize=(8, 6))

sns.heatmap(conf\_matrix, annot=True, fmt="d", cmap="Blues", cbar=False,

            xticklabels=['Non-Fraud', 'Fraud'], yticklabels=['Non-Fraud', 'Fraud'])

plt.xlabel("Predicted Labels")

plt.ylabel("True Labels")

plt.title("Confusion Matrix for Random Forest Model")

plt.show()

**Decision tree Implementation**

import pandas as pd

import numpy as np

import time

from sklearn.model\_selection import train\_test\_split

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

from sklearn.preprocessing import StandardScaler

import matplotlib.pyplot as plt

import seaborn as sns

# Load and preprocess the data

try:

    data = pd.read\_csv('/content/Fraud transaction.csv')

    print("TC-001: Data ingestion - Pass")

except Exception as e:

    print("TC-001: Data ingestion - Fail")

    print(f"Error: {e}")

# Dropping ID column if it exists

data = data.drop(columns=['id'], errors='ignore')

# Define features and target

X = data.drop(columns=['Class'])

y = data['Class']

# Split data into train and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Scaling features

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Initialize and train the Decision Tree model

dt\_model = DecisionTreeClassifier(random\_state=42)

# Test Case TC-002: Verify processing time (simulating real-time processing)

start\_time = time.time()

dt\_model.fit(X\_train, y\_train)

end\_time = time.time()

processing\_time = end\_time - start\_time

if processing\_time < 1:

    print("TC-002: Real-time processing - Pass")

else:

    print("TC-002: Real-time processing - Fail")

# Predictions and model evaluation

y\_pred = dt\_model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

# Test Case TC-003: Model accuracy threshold

if accuracy > 0.95:

    print("TC-003: Fraud detection model accuracy - Pass")

else:

    print("TC-003: Fraud detection model accuracy - Fail")

# Test Case TC-004: False positive rate threshold

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

tn, fp, fn, tp = conf\_matrix.ravel()

false\_positive\_rate = fp / (fp + tn)

if false\_positive\_rate < 0.05:

    print("TC-004: False positive rate - Pass")

else:

    print("TC-004: False positive rate - Fail")

# Print additional results for verification

print("\nModel Accuracy:", accuracy)

print("Classification Report:\n", classification\_report(y\_test, y\_pred))

print("Confusion Matrix:\n", conf\_matrix)

print("False Positive Rate:", false\_positive\_rate)

# Visualize test results with a bar chart

test\_results = {

    'TC-001: Data ingestion': 'Pass' if not data.empty else 'Fail',

    'TC-002: Real-time processing': 'Pass' if processing\_time < 1 else 'Fail',

    'TC-003: Model accuracy > 95%': 'Pass' if accuracy > 0.95 else 'Fail',

    'TC-004: False positive rate < 5%': 'Pass' if false\_positive\_rate < 0.05 else 'Fail'

}

labels = list(test\_results.keys())

results = [1 if result == 'Pass' else 0 for result in test\_results.values()]

colors = ['green' if result == 1 else 'red' for result in results]

plt.figure(figsize=(10, 6))

plt.barh(labels, results, color=colors)

plt.xlabel("Test Result (Pass=1, Fail=0)")

plt.title("Test Case Results for Decision Tree Model")

for i, v in enumerate(results):

    plt.text(v - 0.03, i, "Pass" if v == 1 else "Fail", color="white", va="center", fontweight="bold")

plt.xlim(-0.5, 1.5)

plt.show()

# Visualize the confusion matrix with a heatmap

plt.figure(figsize=(8, 6))

sns.heatmap(conf\_matrix, annot=True, fmt="d", cmap="Blues", cbar=False,

            xticklabels=['Non-Fraud', 'Fraud'], yticklabels=['Non-Fraud', 'Fraud'])

plt.xlabel("Predicted Labels")

plt.ylabel("True Labels")

plt.title("Confusion Matrix for Decision Tree Model")

plt.show()

**Graph Code:**

import pandas as pd

import numpy as np

import time

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.neighbors import KNeighborsClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score, confusion\_matrix, precision\_score, recall\_score, f1\_score

from sklearn.preprocessing import StandardScaler

# Load and preprocess the data

try:

    data = pd.read\_csv('/content/Fraud transaction.csv')

    print("TC-001: Data ingestion - Pass")

except Exception as e:

    print("TC-001: Data ingestion - Fail")

    print(f"Error: {e}")

# Dropping ID column if it exists

data = data.drop(columns=['id'], errors='ignore')

# Define features and target

X = data.drop(columns=['Class'])

y = data['Class']

# Split data into train and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Scaling features

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Initialize models

models = {

    "Logistic Regression": LogisticRegression(max\_iter=10000, random\_state=42),

    "K-Nearest Neighbors": KNeighborsClassifier(n\_neighbors=5),

    "Random Forest": RandomForestClassifier(n\_estimators=100, random\_state=42),

    "Decision Tree": DecisionTreeClassifier(random\_state=42)

}

# Metrics dictionary to store evaluation results for each model

metrics = {

    "Model": [],

    "Accuracy": [],

    "False Positive Rate": [],

    "Processing Time (s)": [],

    "Precision": [],

    "Recall": [],

    "F1 Score": []

}

# Evaluate each model and store metrics

for model\_name, model in models.items():

    print(f"\nEvaluating {model\_name}")

    # Measure training time

    start\_time = time.time()

    model.fit(X\_train, y\_train)

    end\_time = time.time()

    processing\_time = end\_time - start\_time

    # Predictions and metrics

    y\_pred = model.predict(X\_test)

    accuracy = accuracy\_score(y\_test, y\_pred)

    precision = precision\_score(y\_test, y\_pred)

    recall = recall\_score(y\_test, y\_pred)

    f1 = f1\_score(y\_test, y\_pred)

    conf\_matrix = confusion\_matrix(y\_test, y\_pred)

    tn, fp, fn, tp = conf\_matrix.ravel()

    false\_positive\_rate = fp / (fp + tn)

    # Append metrics to dictionary

    metrics["Model"].append(model\_name)

    metrics["Accuracy"].append(accuracy)

    metrics["False Positive Rate"].append(false\_positive\_rate)

    metrics["Processing Time (s)"].append(processing\_time)

    metrics["Precision"].append(precision)

    metrics["Recall"].append(recall)

    metrics["F1 Score"].append(f1)

# Convert metrics to DataFrame

metrics\_df = pd.DataFrame(metrics)

# Melt the DataFrame for grouped bar plotting

metrics\_melted = metrics\_df.melt(id\_vars="Model",

                                 value\_vars=["Accuracy", "False Positive Rate", "Processing Time (s)",

                                             "Precision", "Recall", "F1 Score"],

                                 var\_name="Metric",

                                 value\_name="Score")

# Plot all metrics in a single grouped bar plot

plt.figure(figsize=(14, 8))

sns.barplot(x="Model", y="Score", hue="Metric", data=metrics\_melted, palette="viridis")

plt.title("Comparison of Model Performance Metrics")

plt.ylim(0, 1)  # Adjust this if processing time is much larger than 1

plt.xticks(rotation=45)

plt.legend(loc="upper right")

plt.tight\_layout()

plt.show()

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.NO | Model’s | Accuracy | Precision | Recall | F1\_score |
| 1 | Logistic regression | 69 | 0.12  0.72 | 0.02  0.94 | 0.04  0.81 |
| 2 | K-Nearest Neighbor | 71.3 | 0.43  0.74 | 0.15  0.93 | 0.22  0.82 |
| 3 | Random forest | 70 | 0.30  0.73 | 0.07  0.94 | 0.12  0.82 |
| 4 | Decision tree | 59.3 | 0.27  0.73 | 0.29  0.71 | 0.28  0.72 |

**5.Results and Discussion**

This segment presents the comes about of the card extortion discovery framework, analyzes its viability, impediments and potential regions for enhancement. The talk is based on extortion discovery show execution measurements and real-time preparing capabilities watched amid testing.

**5.1Description**  
This subsection gives a careful examination of the system's execution in identifying wrong trades. Key measurements such as precision, accuracy, survey, disarray network, and F1 score are utilized to evaluate the model's victory in recognizing shakedown without falsely labeling bona fide trades. In expansion, sit out of gear times and reaction times are talked about to assess Apache Flink's real-time preparing capabilities inside the framework**.**

**Key discourse focuses include:**

Illustrate Execution: Surveys how absolutely the illustrate isolates between genuine blue and untrue transactions.  
- Unfaithful Positives and Negatives: Analyzes events where honest to goodness trades were hailed as blackmail (off-base positives) and untrue trades were missed (off-base negatives)- Notice Viability: Examines how rapidly and dependably cardholders are informed of suspicious activity.

- Framework Idleness: Surveys the time taken from information ingestion to extortion discovery, guaranteeing it meets real-time preparation requirements.

**5.2 Graphs**

This area incorporates graphical representations of the system's execution to give a clear, visual understanding of the comes about. A few recommended charts include:

1. Accuracy: Measures the overall correctness of the model by calculating the ratio of correctly predicted instances to the total instances.

Accuracy=TP+TN/TP+TN+FP+FN

2. Precision: Indicates the proportion of positive identifications that were actually correct (True Positives out of all predicted positives).

Precision=TP/TP+FP

3.Recall (Sensitivity): Measures the model's ability to correctly identify all relevant cases (True Positives out of all actual positives).

Recall = TP/TP+FN

4. F1\_Score : The harmonic mean of precision and recall, providing a single metric that balances both.

F1\_score = 2\*precision\*recall/precision + recall.

5. Confusion Matrix : A table layout that summarizes the performance of a classification model, displaying counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

**Comparison of model performance metrics between models**

A graph of different colored bars

Description automatically generated

**6. Testing**

The testing phase makes sure the card fraud detection system meets the established requirements and operates as intended. Test cases that validate every system component from data ingestion to fraud detection, notifications, and dashboard visualization are presented in this section along with an explanation of the testing procedure.

**6.1 Description**

* This subsection describes the overall approach and technique used to test the framework. Testing has been done to guarantee the unwavering quality, accuracy and execution of each module in newly created and real-time situations. The testing process included:
* Unit testing: We tested individual components such as the threat detection demo, the real-time processing module, and the notification framework to ensure that they work correctly individually.
* Integration testing: We ensured that all modules, from data ingestion to threat detection, alerting and visualization, work together consistently. -Platter Test: Examined the ability of a system to handle the highly disproportionate and time-consuming exchanges that warrant a highly disproportionate method.
* Customer Acknowledgment (UAT): Ease of use of the system, especially the notification and visualization modules have been accumulated on the partner side.

**6.2 Test Cases**

**This section provides a few example test cases for each module. Each test case includes the Test ID, Description, Expected Result, and Actual Result.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Description | Expected result | Actual Result | Status |
| TC-001 | Test data ingestion from payment sources. | Data is ingested correctly from all sources and formatted as expected. | Pass. | Pass |
| Tc -002 | Verify real-time processing with apache flink. | Transaction data is processed within acceptable latency(<1 sec per transaction). | Pass. | Pass |
| Tc -003 | Fraud detection model accuracy. | Model correctly classifies transactions with >95% accuracy. | Fail. | Fail. |
| Tc-004 | False positive rate | Model’s false positive rate is <5% | Fail. | Fail. |

**7. Conclusion and Future Enhancements**

This section summarizes the overall performance and effectiveness of the card fraud detection system, highlights its advantages, limitations, and practical applications, and suggests future improvements.

**7.1Conclusion**  
The Card Extortion Locator framework uses machine learning and Apache Flink to efficiently convey information and effectively recognize and flag potentially fake transactions in real time. The framework enables budget education to effectively protect cardholders through accurate classification, tamper notification, and visualization of exchange information. This measure illustrates the potential of real-time analytics and artificial intelligence to improve security and build customer confidence in computerized installment payment systems.

**7.2 Limitations of the Project**

There are also some limitations to the blackmail placement even though the framework provides great possibilities for placing it:

* Have recourse to prepared information: This is due to the fact that impairment of blackmail and substantively proving such blackmail can only be done if there is high quality and a broad swath of prepared information sources. Short or one sided information on factors may result in improper classification.
* Incorrect categorization: With the improvement of models, a small number, if any, of the actual trades would still be regarded as wrongful trades, which may constitute unnecessary friction for the user.
* High exchange: Even though the optimal framework will encounter some stress while dealing with exceptionally high exchange rates, it is not likely gross amounts will result in noticeable delays.
* Limited adaptability: The framework can become obsolete and would have to make further and further modification retraining visits each time new fraud designs appear.

**7.3 Advantages of the Project**

The card extortion tracking framework offers several advantages:

* Real-time detection: The framework can analyze and recognize extortion in seconds, thus providing quick responses.
* Security advancements: It strengthens the security of advanced payment structures, thus reducing the risk of extortion.
* Upgraded Client Believe: Opportune notices offer assistance clients feel more secure around their budgetary transactions.
* Information Visualization: The dashboard highlight gives valuable bits of knowledge for both checking and future analysis.

**7.4 Applications of the Project**

This framework has wide application in various sectors related to money and retail:

* Cash custody and budget administration: Real-time extortion localization can be coordinated with cash storage applications to protect customers from unauthorized transactions.
* E-commerce stages: It is important for online retailers to verify and protect installment exchanges, thereby reducing losses due to fraud.
* Exchange of automatic ticket distributors: verification of automatic distributors of real -time tickets may provide for the unauthorized withdrawal of funds.
* Protection: can be adjusted to distinguish false allegations, analyzing the design of exchange and inconsistencies.

**7.5 Future Enhancements**

Several changes can be made to improve the system:

* Universal machine learning models: Build models that can continuously learn and adapt to growing black mail designs without the need form anual retraining.
* Improved points of inconsistency: Coordinated advanced inconsistency detection strategies to recognize complex extortion design sand minimize false positives.
* Multi-factor authentication integration: Include multi-factor authentication features on supported exchanges for increased security.
* Prescient Analytics: Use prescient analytics to proactively identify proposals that may have recently exhibited a risk of extortion.
* Blockchain Integration: Explore the use of blockchain innovation to confirm exchange and include an additional layer of security.

**REFERENCES & BIBLIOGRAPHY**

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

This paper provides an overview of statistical approaches for fraud detection, examining methods such as anomaly detection and supervised learning.

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

- This review explores various data mining techniques applied in financial fraud detection, including machine learning methods like neural networks and decision trees.

1. 3Zhang, Y., & Zhou, X. (2018). A comparative study of fraud detection techniques for financial data. International Journal of Computer Applications, 975-8887.

- This study compares different fraud detection techniques, such as rule-based, machine learning, and deep learning approaches, to determine their effectiveness in financial datasets.

1. 4. Chen, H., Chung, W., Xu, J. J., Wang, G., Qin, Y., & Chau, M. (2004). Crime data mining: A general framework and some examples.Computer, 37(4), 50-56.

- Although focused broadly on crime data mining, this paper provides useful insights into data mining approaches applicable to financial fraud detection

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

- This study compares several data mining algorithms, including logistic regression, decision trees, and neural networks, in the context of credit card fraud detection.

1. Flink Documentation - Real-time stream processing with Apache Flink.(n.d.). Retrieved from [Apache Flink Documentation](https://flink.apache.org/)

- Official documentation for Apache Flink, providing insights on real-time data processing, which is critical in real-time fraud detection scenarios.

1. Wei, W., & Wang, Z. (2018). Credit card fraud detection using convolutional neural networks. Journal of Cyber Security Technology, 2(3-4), 67-74.

- This paper explores the use of deep learning, specifically convolutional neural networks (CNNs), for detecting credit card fraud.

1. Whitrow, C., Hand, D. J., Juszczak, P., Weston, D., & Adams, N. M.(2009). Transaction aggregation as a strategy for credit card fraud detection. Data Mining and Knowledge Discovery, 18(1), 30-55.

- Discusses the importance of aggregating transaction data to detect fraud

patterns that may not be evident in single transactions.

1. Kirkos, E., Spathis, C., & Manolopoulos, Y. (2007). Data mining techniques for the detection of fraudulent financial statements. Expert Systems with Applications, 32(4), 995-1003.

- This paper examines the use of data mining techniques, such as decision trees and neural networks, for detecting fraud in financial statement data.

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

- A comprehensive review of fraud detection techniques in the context of credit card fraud, exploring both rule-based and machine learning approaches.