Intelligent Real-Time Fraud Detection in Banking Systems Using Machine Learning

Rohini N[1], D. Murali[2]
[1]PG Student, QUBA College of engineering and technology
[2]Associate Professor, QUBA College of engineering and technology

Abstract— **Financial fraud refers to the act of gaining monetary benefits through dishonest and illegal practices. In recent years, it has emerged as a major threat to businesses and organizations. Despite multiple efforts aimed at mitigating financial fraud, it continues to inflict significant damage on both the economy and society, with daily financial losses reaching substantial levels. Initial fraud detection techniques were introduced several years ago; however, most traditional approaches were manual in nature, making them inefficient, costly, and prone to inaccuracies. Although ongoing research attempts to develop better solutions, the problem of financial fraud remains largely unresolved. Traditional methods, such as manual verifications and audits, are often inaccurate, labor-intensive, and expensive. With the advancement of artificial intelligence, it is now possible to leverage machine learning techniques to analyze large volumes of financial data and effectively detect fraudulent activities. Therefore, this study proposes a new model for fraud detection in bank transactions, utilizing the Random Forest Classifier Machine Learning Algorithm. Using the Banksim dataset, our model demonstrates superior performance compared to existing systems, achieving 99% accuracy in both training and testing phases.**

Index Terms— **Financial Fraud, Dishonest Financial Gains, Unlawful Means, Business Threats**

I. INTRODUCTION

The increasing prevalence of digital banking and online transactions has made fraud detection a critical component of banking operations. Financial crimes such as identity theft, account takeovers (ATO), and credit card fraud can result in significant financial losses for both institutions and their clients. Traditionally, fraud detection has relied on rule-based systems, which often fall short in identifying emerging and sophisticated fraudulent schemes. In contrast, machine learning (ML) provides a more advanced and precise approach by analyzing large volumes of data to uncover patterns indicative of fraudulent activity. The integration of artificial intelligence (AI) has further transformed fraud detection in internet banking, with many organizations now incorporating AI-driven analytics into their fraud prevention systems. ML and AI technologies are capable of rapidly processing extensive datasets to identify unauthorized transactions and suspicious behavior. By leveraging these technologies, financial institutions can enhance their ability to detect and prevent fraud, thereby safeguarding their customers' assets more effectively. For businesses aiming to expand fraud detection measures within online banking and fintech, it is essential to understand how machine intelligence (MI) can be utilized and why it offers significant advantages over traditional detection methods.

II.LITERATURE SURVEY

Authors: S. Delecourt and L. Guo
Description: Mobile payment systems are increasingly becoming a dominant method of transaction across numerous countries. Nevertheless, the incidence of fraud associated with mobile payments is notably higher compared to traditional credit card usage. A contributing factor to this trend is the relative ease with which mobile data can be manipulated by fraudsters, thereby weakening the effectiveness of data-driven fraud detection mechanisms. While supervised learning techniques are widely utilized in fraud detection, they are typically designed under the assumption of a benign environment, where no active adversaries are attempting to circumvent the system. In this study, we incorporated the potential countermeasures taken by fraudsters into the development of a more resilient mobile fraud detection framework, employing adversarial examples. The

experimental outcomes demonstrated that our proposed approach enhanced system performance in both benign and adversarial conditions

II. IMPORTANCE OF SMART METERS DATA PROCESSING CASE OF SAUDI ARABIA, AUTHORS: T. ALUTHGAMA, A. M. ALSUBAIE, AND M. ANWER

Description: This study conducts a comprehensive evaluation of 30-minute interval datasets collected from residential digital meters across the Kingdom of Saudi Arabia (KSA). The objective is to identify all possible discrepancies within the datasets and to develop statistical techniques tailored to address each type of discrepancy. The analysis was carried out using a custom-built program developed with Python-Pandas, which processes three months' worth of meter readings from 3,283 consumers across KSA. The program systematically detects various data issues, including inconsistencies, duplicate entries, missing values, outliers, and other anomalies. Following detection, appropriate statistical methods integrated into the program are applied to correct these problems. Additionally, a validation procedure was incorporated to verify that the corrections result in the most accurate and dependable dataset. Findings reveal that smart meter data typically require preprocessing before they can be effectively utilized for further applications. The final outcome demonstrates that, after preprocessing, the smart meter measurement datasets can be deemed valid and reliable for use.

III. SYSTEM ANALYSIS

3.1 Several studies have been conducted to address fraud detection across various sectors:

- Abdallah et al. provided a comprehensive review focused on identifying fraudulent activities in the healthcare domain, primarily using statistical methods. Their work examined traditional statistical approaches and their effectiveness in fraud detection.
- Popat and Chaudhary presented an extensive review of credit card fraud detection techniques. They offered a detailed evaluation of various machine learning (ML) classification algorithms, along with a discussion on methodologies and the inherent challenges associated with each.
- Ryman-Tubb et al. analysed multiple state-of-the-art techniques for detecting fraudulent activities in payment card transactions, particularly considering large-scale transactional volumes. Their findings revealed that only a limited number. Eight approaches. Show real potential for practical application within industry environments.
- Albashrawi and Lowell investigated a decade's worth of studies on fraud detection in the financial sector using data mining techniques. However, their review lacked comprehensiveness, failing to adequately discuss evaluation methods and neglecting the advantages and disadvantages associated with the various data mining approaches.

Disadvantages of Existing Systems

Despite significant advancements, current fraud detection systems exhibit several limitations:

- Systems utilizing Logistic Regression models struggle to predict continuous outcomes, limiting their flexibility in certain fraud detection scenarios.
- The performance of Logistic Regression models diminishes considerably with small sample sizes, leading to unreliable predictions.
- Existing systems are highly susceptible to the overfitting problem, capturing noise rather than general patterns within the data.
- The accuracy of the models heavily relies on the quality of the input data; noisy or incomplete data can severely affect results.
- When dealing with large datasets, the prediction process becomes slow and computationally inefficient.
- Logistic Regression models are sensitive to the scaling of features and can be adversely impacted by irrelevant attributes within the data.
- High memory requirements pose another challenge, as these models often necessitate storing the complete training dataset for reference during prediction stages.

3.2.1. Advantages:

- It will give better accuracy
- Better accuracy

3.3. System Requirements

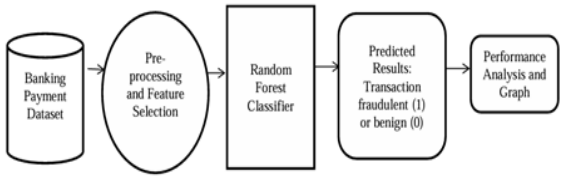
3.3.1. HARDWARE REQUIREMENTS (minimum)

- System : Pentium IV 2.4 GHz
- Hard Disk : 40 GB
- Ram : 512 Mb. 3.3.2.

Software Requirements

- Operating System: Windows
- Coding Language: Python 3.7

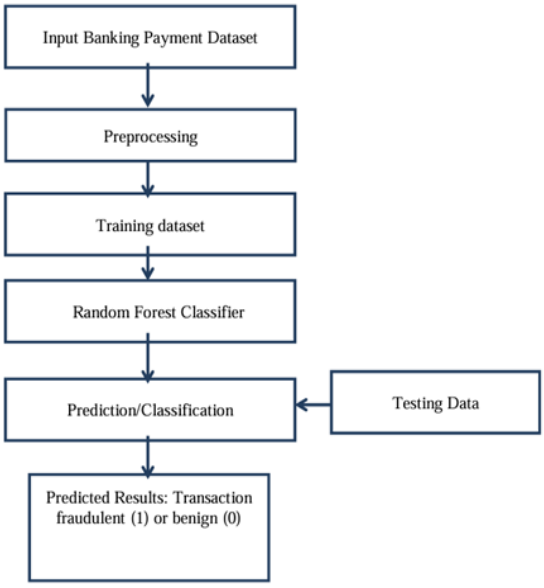
IV. SYSTEM ARCHITECTURE



V. SYSTEM DESIGN

Data Flow Diagram (DFD)

1. The Data Flow Diagram (DFD), also known as a bubble chart, is a simple yet powerful graphical tool used to represent a system. It illustrates the flow of input data into the system, the processing performed on that data, and the resulting output data produced by the system.
2. DFDs serve as one of the most critical modelling tools in system analysis and design. They are used to model the core components of a system, including the processes within the system, the data consumed and generated by these processes, external entities that interact with the system, and the movement of information throughout the system.
3. A DFD captures how data moves within the system and the transformations it undergoes as it flows from input to output. This graphical technique highlights the flow of information and the operations performed during different stages of processing.
4. Often referred to as a bubble chart, a DFD can represent a system at varying levels of abstraction. It can be partitioned into multiple levels, where each level provides greater detail regarding the information flow and the functional aspects of the system.



VI. SOFTWARE ENVIRONMENT

Before diving into the various methods of machine learning, it's helpful to first understand what machine learning is and what it isn't. Often, machine learning is categorized as a subset of artificial intelligence (AI), but this classification can sometimes be misleading. While machine learning did emerge from AI research, in the context of data science, it's more accurate to think of machine learning as a tool for constructing models of data.

At its core, machine learning is about creating mathematical models that help us make sense of data. The term "learning" comes into play when these models are given adjustable parameters, which can be tuned based on data they encounter. This allows the model to adapt and improve over time. Once trained on historical data, these models can then be used to predict or interpret new, unseen data.

The degree to which this type of "learning" resembles the learning process in humans is a matter of philosophical debate. However, it's crucial to understand the problem context in machine learning, as this will enable effective use of these methods. Machine learning is employed in a variety of applications, including image and speech recognition, natural language processing, recommendation systems, fraud detection, portfolio optimization, and automated tasks. Additionally, machine learning algorithms are the driving force behind technologies

like autonomous vehicles, drones, and robots, making them more intelligent and capable of adapting to changing environments.

VII. SYSTEM IMPLEMENTATION

```
mport numpy as np
import pandas as pd
from flask import Flask, request, jsonify, render template, redirect, flash, send_file
from sklearn. preprocessing import MinMaxScaler
from werkzeug. utils import secure_filename
import pickle
from sklearn. tree import DecisionTreeClassifier
from sklearn.svm import SVC

# Initialize the Flask app
app = Flask(__name__)
# Load pre-trained model
fraud = pickle. Load (open ('fraud.pkl', 'rb'))
@app. route ('/')
@app. route('/first')
def first ():
return render template('first.html')
if __name__ == "__main__":
apparent(debug=True)
```

VIII. SYSTEM TESTING

System Testing

The primary goal of testing is to uncover errors within a system. It involves systematically attempting to identify any faults or weaknesses in a product. Testing provides a way to evaluate the functionality of individual components, sub-assemblies, complete assemblies, and/or the final product. It is a process of exercising software to ensure that the system meets its specified requirements and user expectations while preventing failures that could lead to unacceptable outcomes. There are several types of tests, each addressing a unique aspect of the system.

Types of Tests

1. Unit Testing

Unit testing focuses on verifying that the internal logic of individual components of the software works correctly. It involves designing test cases that validate whether program inputs produce expected outputs and ensuring all decision branches and internal code flow are correct. Unit testing is performed on individual software units before they are integrated into the larger system. It is a type of structural testing, which requires knowledge of the internal design of the software, and it is considered invasive because it tests the internal workings of each unit.

Unit tests are crucial for verifying that each specific business process, application function, or system configuration works as intended. They ensure that each unique path of a business process adheres to the documented specifications and delivers the expected results with clearly defined inputs.

2. Integration Testing

Integration testing involves testing the interaction between different software components that have already been unit tested to ensure they work together seamlessly. The goal is to determine whether the integrated components function correctly as a unified system. This type of testing is typically event-driven, focusing on the outcomes of interactions, such as the correct display of fields or screens.

Although each component may have passed unit testing successfully, integration testing aims to uncover problems that may arise when components are combined. This helps ensure that the integration of components is correct and consistent, and it addresses issues that might not have been identified during unit testing.

IX SCREENSHOTS

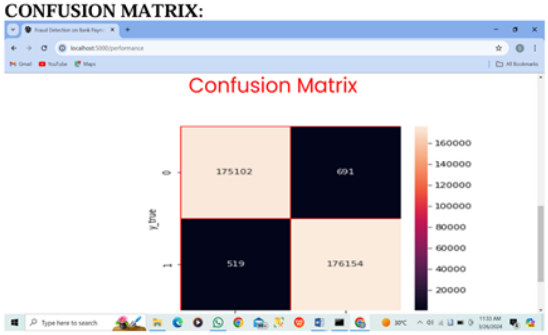
INDEX PAGE:

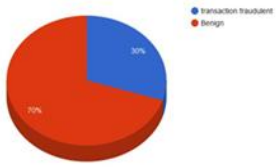
Fig 9.1: Index Page

LOGIN PAGE:

Fig 9.2: Login Page

CONFUSION MATRIX:





X. CONCLUSION

Financial fraud can manifest in various sectors, such as corporate finance, banking, insurance, and taxation. Recently, financial fraud has become a growing concern for businesses and industries alike. Despite numerous efforts to combat it, financial fraud persists, causing significant damage to both society and the economy. The daily financial losses due to fraud are staggering and continue to have a profound impact.

With advancements in artificial intelligence, machine learning (ML)-based technologies now offer a promising solution to identifying fraudulent transactions. By analysing large volumes of financial data, these technologies can intelligently detect anomalies that may indicate fraudulent activity.

This study presents a comprehensive analysis and synthesis of existing knowledge on ML-based fraud detection methods. We specifically focus on the use of the Random Forest Classifier, which employs well-defined techniques to extract insights, synthesize data, and report findings effectively

REFERENCES

[1] S. Delecourt and L. Guo, "Building a robust mobile payment fraud detection system with adversarial examples," in 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE) pp. 103–106, IEEE, 2019.

[2] T. Aluthgama, A. M. Alsubaie, and M. Anwer, "Importance of smart meters data processing – case of Saudi Arabia," in 2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA), pp. 1–5, IEEE, 2019.

[3] O. Adepoju, J. Wosowei, S. Lawte, and H. Jaiman, "Comparative evaluation of credit card fraud detection using machine learning techniques," in 2019 Global Conference for Advancement in Technology (GCAT), pp. 1– 6, IEEE, 2019.

[4] S. Khatri, A. Arora, and A. P. Agrawal, "Supervised machine learning algorithms for credit card fraud detection: A comparison," in 2020 10th International Conference on Cloud Computing, Data Science Engineering (Confluence), pp. 680–683, IEEE, 2020.

[5] V. Jain, M. Agrawal, and A. Kumar, "Performance analysis of machine learning algorithms in credit cards fraud detection," in 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), pp. 86 88, IEEE, 2020.

[6] Tennakoon, C. Bhagyani, S. Premadasa, S. Mihiranga, and N. Ku- ruwitaarachchi, "Real-time credit card fraud detection using machine learning," in 2019 9th International Conference on Cloud Computing, Data Science Engineering (Confluence), pp. 488–493, IEEE, 2019.