A Major Project Final Report on

Transaction Integrity Verification System (TIVS)

Submitted in Partial Fulfillment of the
Requirements for the Degree of
Bachelor of Engineering in Sofware Engineering
under Pokhara University

Submitted by:
Aananda Bhusal, 191701
Sachit Khadka, 191722
Sampada Kharel, 191723
Sibendra Timalsina, 191727
Subina Maharjan, 191734

Under the supervision of:
Er. Subash Manandhar

Date:
27 JULY 2024

Department of Software Engineering
NEPAL COLLEGE OF
INFORMATION TECHNOLOGY
Balkumari, Lalitpur, Nepal

ACKNOWLEDGEMENT

We extend our heartfelt gratitude to the college, dedicated teachers, and the esteemed faculty members of the Software Engineering department. Their collaborative efforts paved the way for the successful realization of this project. Their unwavering support and guidance were instrumental in the success of this project.

Special thanks go to our Project Supervisor **Er. Subash Manandhar** for his invaluable guidance and technical expertise, which played a crucial role in the development and growth of our project. His insightful suggestions propelled our project to new heights.

We are also indebted to our teachers, parents, and colleagues for their contributions, whether known or unknown. Their support and valuable insights were a cornerstone of this project's development. Thank you all for being an integral part of this journey.

Aananda Bhusal, 191701

Sachit Khadka, 191722

Sampada Kharel, 191723

Sibendra Timalsina, 191727

Subina Maharjan, 191734

ABSTRACT

The Transaction Integration Verification System (TIVS) is designed to enhance the security and reliability of online financial transactions. TIVS processes transactional data through three critical steps to ensure thorough analysis and anomaly detection. First, transactions are checked against a blacklist to identify accounts with prior issues. Next, the data passes through a rules engine, applying various rules defined by financial institutions and assigning positive or negative weights accordingly. Finally, AI algorithms predict any anomalies within the transactions. By continuously analyzing transactional data in real-time, TIVS effectively detects fraudulent activities and potential money laundering scenarios. Its capability to swiftly identify unusual trends and alert financial institutions and regulatory bodies ensures timely action and maintains the integrity of online financial transactions.

Keywords: *Artificial Intelligence, Fraud Detection, Money Laundering Prevention, Real-Time Monitoring, Transaction Security*

TABLE OF CONTENTS

List of Figures

List of Abbrevations

TIVS	Transaction Integrity Verification System
AI	Artificial Intelligence
ML	Machine Learning
AML	Anti-Money Laundering
BAF	Bank Account Fraud
LightGBM	Light Gradient Boosting Machine
GOSS	Gradient based One Side Sampling
EFB	Exclusive Feature Bundling
UI	User Interface
UX	User Experience

1. INTRODUCTION

The Transaction Integration Verification System (TIVS) is a cutting-edge tool designed to enhance the security and reliability of online financial transactions. As online transactions become increasingly prevalent, the need for robust systems to detect and prevent fraudulent activities and money laundering is more critical than ever. TIVS addresses this need by processing transactional data through three critical steps to ensure thorough analysis and anomaly detection.

First, TIVS checks transactions against a blacklist to identify accounts with prior issues, ensuring that known threats are promptly flagged. Next, the data passes through a sophisticated rules engine, where various rules defined by financial institutions are applied, assigning positive or negative weights accordingly. This step ensures that the system adapts to the specific risk profiles and regulatory requirements of different financial institutions. Finally, advanced AI algorithms predict any anomalies within the transactions, providing a dynamic and intelligent layer of security.

By continuously analyzing transactional data in real-time, TIVS effectively detects fraudulent activities and potential money laundering scenarios. Its ability to swiftly identify unusual trends and alert financial institutions and regulatory bodies ensures timely action, thereby maintaining the integrity of online financial transactions. This introduction sets the stage for exploring the detailed workings and benefits of TIVS, highlighting its role in safeguarding the modern financial ecosystem.

1.1 Problem Statement

In the digital age, the increasing reliance on online financial transactions has introduced significant security and integrity challenges. Despite existing safeguards, fraudulent activities and money laundering persist as major threats to the financial system. These threats result in potential financial losses, reputational damage, and regulatory penalties for financial institutions. Moreover, they undermine public trust in the financial system, discourage legitimate business activities, and facilitate other criminal enterprises. Current systems often fail to detect and prevent these threats in real time, highlighting the critical need for enhanced methods to secure online transactions, accurately analyze transactional data, and quickly identify and address fraudulent activities and money laundering attempts.

1.2 Project Objectives

1. To detect and prevent financial fraud and money laundering in real-time.
2. To alert financial institutions and regulatory bodies to take timely action.
3. To provide insights and historical data on fraud detection and money laundering.

1.3 Significance of the study

The Transaction Integrity Verification System (TIVS) plays a vital role in maintaining financial integrity and combating emerging threats in Nepal's evolving financial landscape. By detecting and preventing financial fraud and money laundering in real time, TIVS acts as a safeguard, ensuring the security and reliability of transactions. This protection is crucial for building trust and stability within the financial ecosystem.

TIVS also provides timely alerts to financial institutions and regulatory bodies, enabling them to take swift action against fraudulent activities and money laundering. This proactive approach not only protects the financial interests of individuals and institutions but also upholds the country's credibility and economic progress.

1.4 Scope and Limitations

1.4.1 Scope

1. Detect and prevent fraudulent activities and money laundering in real time.

2. Analyze transactional data in detail, identify odd trends and patterns, and alert financial institutions and regulatory bodies for timely action.

3. Safeguard the integrity of online financial transactions and scale and integrate with existing financial systems.

1.4.2 Limitations

1. The effectiveness of the TIVS depends on the quality and accuracy of the transactional data it receives.

2. If the data is incomplete or inaccurate, it may affect the system's ability to detect fraudulent activities.

3. Additionally, TIVS relies on AI algorithms, which may have limitations in terms of false positives or false negatives.

4. Furthermore, the implementation of the TIVS may require collaboration and cooperation from various financial institutions, regulatory bodies, and technology providers.

2. LITERATURE REVIEW

2.1 Related Theory

1. **Anti-Money Laundering (AML) Detection Using AI/ML:**

   Anti-money laundering refers to the laws, regulations, and procedures implemented to prevent criminals from disguising illegally obtained funds as legitimate income. AML systems employ comprehensive strategies, including customer due diligence, transaction monitoring, and reporting suspicious activities. Advanced AML techniques utilize artificial intelligence and machine learning to enhance the detection and prevention of money laundering schemes by identifying patterns and anomalies in financial transactions.

   Machine Learning (ML) algorithms have been extensively researched for their potential to detect money laundering activities. Decision trees, support vector machines (SVM), and neural networks have been widely applied. For instance, random forests and gradient boosting machines (GBM) have shown efficacy in handling large datasets and identifying complex patterns indicative of money laundering [2].

   XGBoost (Extreme Gradient Boosting) has also gained attention for its superior performance in AML systems. XGBoost is particularly effective in managing high-dimensional data and imbalanced datasets, common in financial transactions. Its ability to incorporate regularization helps prevent overfitting, thereby improving the model's generalization capabilities [3]. Studies have demonstrated that XGBoost can significantly reduce false positives while maintaining high detection rates for suspicious activities [11].

2. **Artificial Intelligence and Machine Learning in Fraud Detection:**

   Automated systems are being developed to detect and measure risks associated with fraud cases, leveraging advanced technologies such as Artificial Intelligence (AI) and Machine Learning (ML). The private insurance sector, in particular, is rapidly adopting these technologies to combat fraudulent transactions. According to Dhieb, Ghazzai, Besbes, and Massoud [5], frameworks utilizing ML models like Gradient Boosting exhibit significantly higher accuracy levels in fraud detection, with an improvement of approximately seven percent compared to traditional models such as decision trees. These frameworks are supported by automated data collection systems, which gather the necessary information to identify fraudulent cases. The extensive research and practical applications in sectors like insurance provide a valuable testing ground for the financial industry to enhance fraud detection capabilities[5].

   The private insurance sector is one area that has rapidly adopted AI and ML technologies to combat fraud. Automated systems within this sector utilize a combination of advanced ML algorithms and comprehensive data collection mechanisms to detect fraudulent claims effectively. These systems gather and analyze relevant data, such as policyholder information, claim history, and transaction details, to identify potential fraud cases. The integration of AI in insurance fraud detection has provided significant improvements in both accuracy and operational efficiency, serving as a valuable testing ground for broader financial industry applications [1].

2.2 Related Work

In our project research, we found that several companies offer fraud detection and AML solutions using AI and machine learning. However, these companies protect their information and implementation methodologies as trade secrets, which limits the availability of detailed academic research on their specific algorithms and techniques. As a result, we could not conduct an in-depth analysis of the specific software. Nonetheless, through extensive research, we were able to extract an overview and some basic approaches of these software solutions.

1. **NICE Actimize overview:** NICE Actimize delivers a comprehensive suite of financial crime, risk, and compliance solutions. Their platform incorporates advanced analytics and machine learning to detect and mitigate fraud and AML risks. Key services include transaction monitoring, case management, and regulatory reporting [10].

2. **FICO overview:**

   FICO provides a variety of fraud detection and AML solutions driven by predictive analytics and machine learning. Their services cover credit card fraud detection, transaction monitoring, and AML compliance, aimed at identifying, investigating, and managing financial crimes [6].

3. **Oracle Financial Services Analytical Applications (OFSAA):**

   OFSAA offers a suite of tools for fraud detection and AML compliance. The platform integrates diverse data sources and applies advanced analytics to monitor transactions, detect anomalies, and maintain regulatory compliance.

4. **DataVisor Overview:**

   DataVisor is an advanced AI-powered fraud detection platform that employs machine learning techniques and a global intelligence network to safeguard businesses from diverse fraud types. The platform integrates proprietary unsupervised machine learning algorithms to identify both known and unknown fraudulent activities without requiring extensive training datasets. It effectively handles multi-channel fraud, including account takeovers, application fraud, payment fraud, and synthetic identity fraud by analyzing digital footprints and behavioral data [4]

The insights gained from the literature review have profoundly influenced the development of our fraud detection and AML system. We adopted several key methodologies and strategies based on the reviewed literature.

Firstly, we incorporated advanced machine learning algorithms, such as gradient boosting and XGBoost, into our system. These algorithms were chosen for their proven effectiveness in handling high-dimensional and imbalanced data, as demonstrated in existing AML systems. This choice was aimed at enhancing detection accuracy and minimizing false positives. Additionally, our system was designed to integrate diverse data sources comprehensively. This approach was inspired by best practices observed in the literature, which emphasized the importance of providing a holistic view of transactions to better identify anomalies and suspicious activities.

We also focused on incorporating real-time analysis and adaptation capabilities. This feature enables our system to dynamically respond to new fraud patterns and emerging threats, ensuring that detection rates remain high across various fraud scenarios.

Finally, we implemented robust mechanisms for regulatory compliance and reporting. By adopting best practices for transaction monitoring and reporting, we ensure that our system meets legal requirements and provides timely alerts to relevant authorities.

By integrating these advanced techniques and strategies, our project aims to create a robust and effective fraud detection and AML system that leverages the latest advancements in AI and machine learning to comprehensively address financial crimes.

3. METHODOLOGY

3.1 Incremental Software Process Model

We’ll adopt the Incremental Software Development Model, tackling requirements, research, design, training, evaluation, and documentation in iterative cycles.

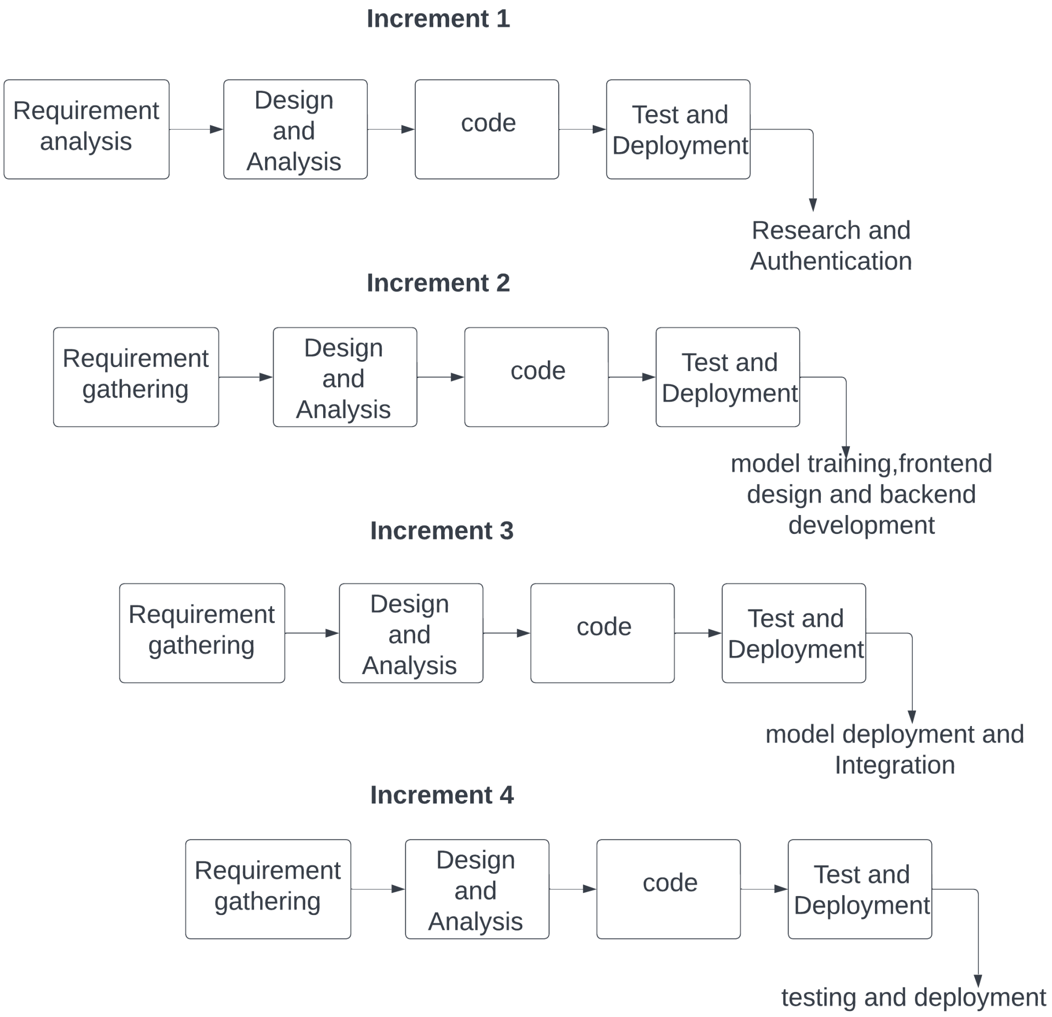


Figure 1: Incremental Model

1. **First Increment:**
    - **Research and Setup:**
        - Conducted thorough research on algorithms and related work to inform the project's direction.
        - Set up the overall project structure, laying a strong foundation for subsequent development.
    - **User Authentication:**
        - Built a user-friendly user authentication system, including login and sign-up pages.
        - Implemented proper validation to ensure secure and accurate user authentication.
        - Tested the functionality and validation of the authentication system to confirm reliability.
2. **Second Increment:**
    - **Model Training and Frontend Development:**
        - Selected a suitable algorithm for model training.
        - Trained the Anti-Money Laundering (AML) and bank fraud detection models.
        - Developed the frontend design, incorporating features such as login-signup pages, dashboards, and insights.
    - **Backend Development:**
        - Implemented the authentication system in the backend.
        - Developed logic to detect blacklisted users or accounts, canceling their transactions immediately and reporting to au-

thorities.

  - Created a rules engine to calculate the weight of rules positively and negatively mentioned by the bank. Transactions failing to meet the defined threshold are automatically terminated.

3. **Third Increment:**

   - **Enhanced Model Training and Integration:**

     - Conducted further model training to improve accuracy.
     - Integrated the AML and bank fraud detection models with the existing system.

   - **System Integration:**

     - Connected the backend with the frontend, ensuring seamless data flow and interaction.
     - Implemented a live dashboard on the frontend, providing real-time insights and visualizations of the data.
     - Conducted rigorous testing to verify the functionality and reliability of the entire system.

4. **Fourth Increment:**

   - **System Testing and Deployment:**

     - Performed comprehensive system integration testing to ensure all components work harmoniously.
     - Deployed the fully integrated system, making it operational for end-users.

3.2 Use case Diagram

A use case diagram is a visual representation that illustrates the interactions between users (actors) and a system, depicting the various ways users interact with the system to achieve specific goals or tasks.

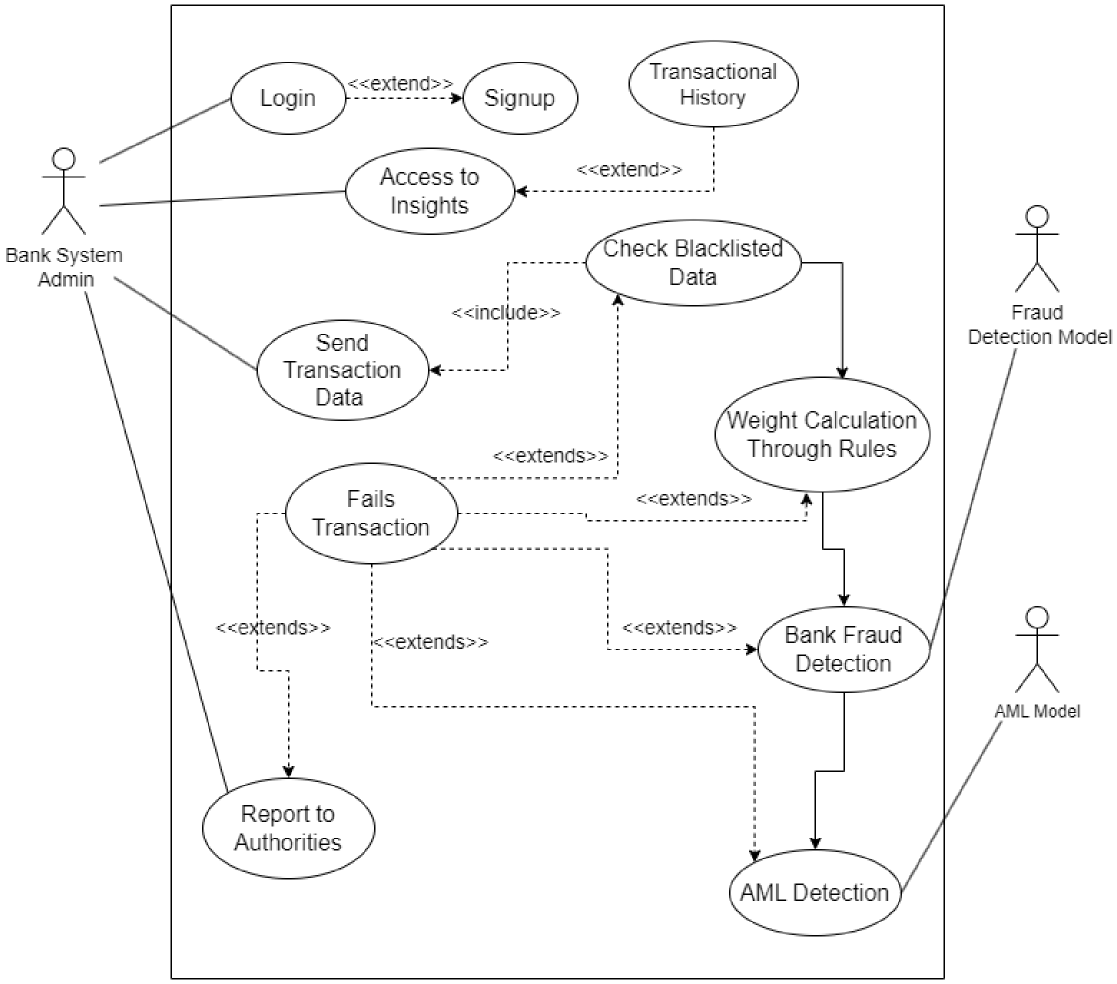


Figure 2: Use Case Diagram

Our system involves the Bank System Admin, Fraud Detection Model, and Anti-Money Laundering (AML) Model working together to ensure transaction security. The process starts with the Bank System Admin logging in, followed by the bank sending transaction data. This data is first checked against blacklisted users, then evaluated by the Rules Engine. If the transaction passes these steps, it undergoes fraud detection and AML analysis. Suspicious or risky transactions are reported to authorities. Successful transactions are presented to the admin through insightful graphs, ensuring thorough scrutiny and transparency.

3.3 Sequence Diagram

A sequence diagram is a flowchart-like illustration that shows how different parts of a system interact with each other over time. Imagine vertical lines representing participants (like user apps or services) and horizontal arrows depicting messages exchanged between them. As time goes down the page, the sequence of messages becomes clear, helping visualize how a process unfolds.

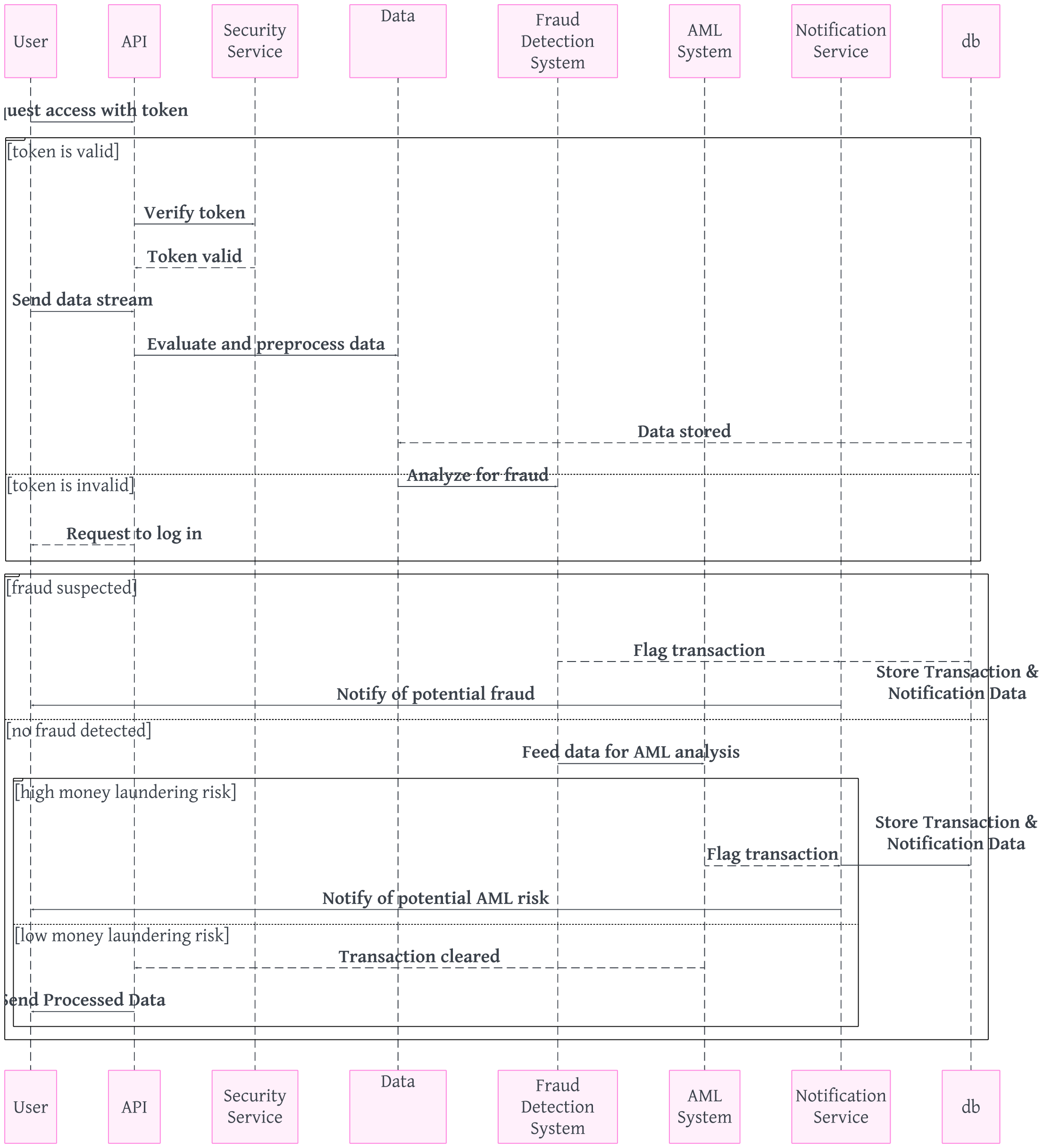


Figure 3: Sequence Diagram

First, it checks your access with a security token. If valid, the data goes through cleaning and analysis. Here, two models come into play: fraud detection and anti-money laundering (AML).

The fraud detection model searches for suspicious activity. If it finds something fishy, the transaction gets flagged and relevant people are notified. If not, the AML model takes over, looking for patterns that might indicate money laundering. Again, if the risk is high, the transaction is flagged and people are notified.

Finally, if everything seems okay, the processed data goes back to you. This system, with its security checks, data cleaning, and specialized models, helps guard against financial crimes.

3.4 Flowchart

A flowchart is a visual representation of a process, outlining the steps involved, decisions made, and the flow of information or data. It uses standardized shapes like rectangles (processes), diamonds (decisions), ovals (start/end points), and arrows (flow direction) to create a clear and easy-to-understand diagram.

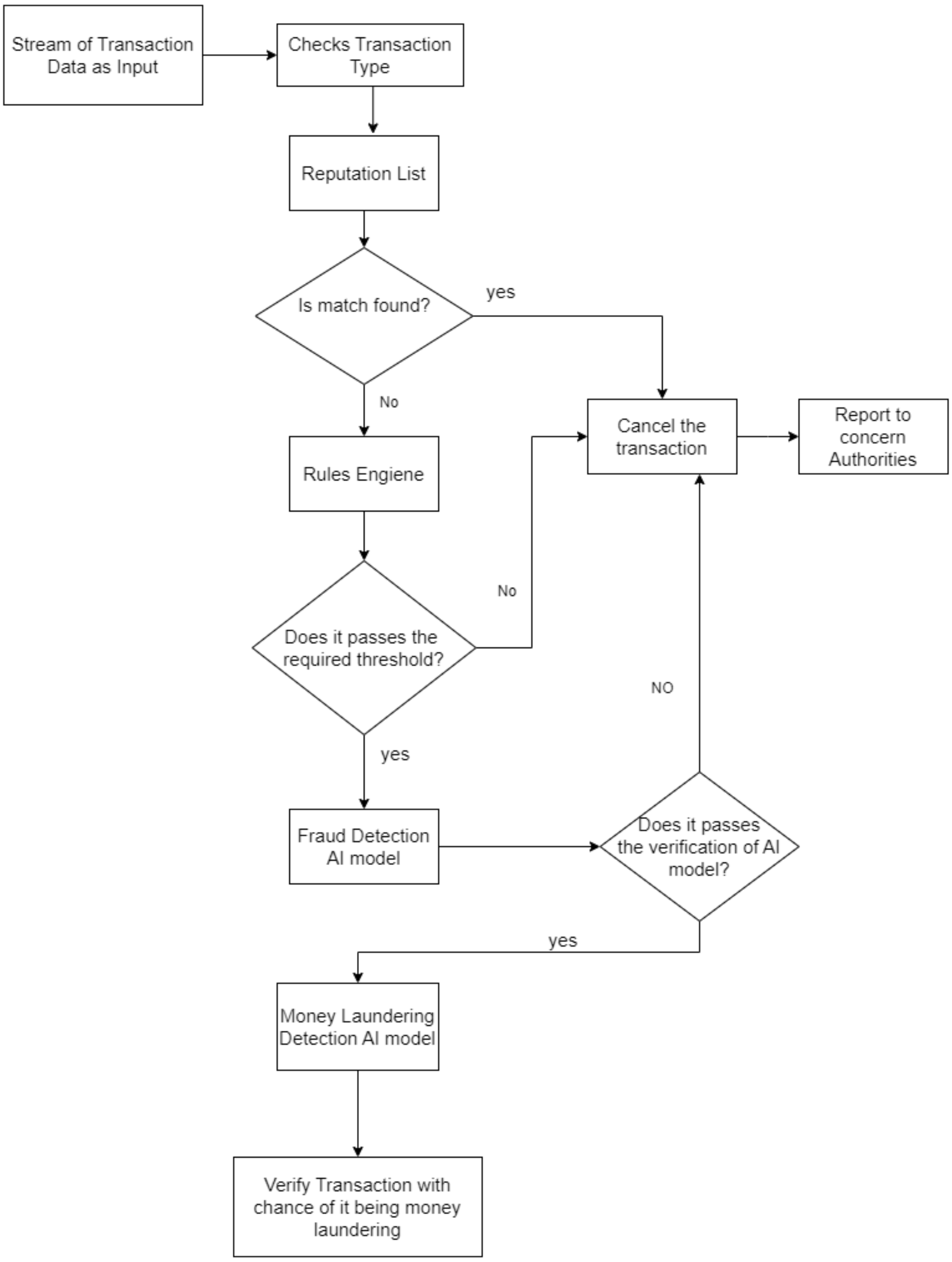

Stream of Transaction Data as Input
Checks Transaction Type
Reputation List
Is match found?
yes
No
Cancel the transaction
Report to concern Authorities
Rules Engiene
No
Does it passes the required threshold?
NO
yes
Fraud Detection AI model
Does it passes the verification of AI model?
yes
Money Laundering Detection AI model
Verify Transaction with chance of it being money laundering

Figure 4: Flowchart

The flowchart represents the process of handling user data, specifically financial transactions, and assessing them for potential fraud and money laundering risks. The process consists of several key components: Reputation List, Rules Engine, Bank Fraud Detection AI Model, and Money Laundering Prediction AI Model. Below is a detailed explanation of each step involved in the system:

1. **Transaction Initiation**:
   - The process begins when a user initiates a transaction. The transaction data is streamed into the system for analysis.
   - Three types of transactions are evaluated in the system: Credit card transactions, E-commerce transactions, and Bank transactions.
2. **Reputation List Checking**:
   - The first component the transaction data encounters is the Reputation List.
   - The system evaluates the transaction information against a list of previously blacklisted users.
   - If a match is found, indicating that the transaction involves a blacklisted user, the transaction is immediately canceled.
   - If no match is found, the transaction data proceeds to the next step.
3. **Rules Engine Evaluation**:
   - In this step, the transaction data is assessed by the Rules Engine.
   - Rules are different for different types of transactions, so the system evaluates the rules according to the transaction type.
   - The Rules Engine calculates the weighted rules (both positive and negative) are predefined in existing financial system .

- The transaction is evaluated against these rules to determine if it meets the defined threshold.
- If the transaction does not meet the threshold, it is canceled.
- If the transaction meets the threshold, it proceeds to the next step.

4. **Fraud Detection AI Model**:
    - The fraud detection AI model consists of three sub-models: Credit card fraud detection AI model, E-commerce fraud detection AI model, and Bank transaction fraud detection AI model.
    - According to the transaction data type, the data enters the respective AI model.
    - These AI models are trained to detect fraudulent activities based on transaction patterns.
    - If the model identifies the transaction as fraudulent, the transaction is canceled.
    - If the model considers the transaction safe, it proceeds to the next step.
5. **Anti-Money Laundering (AML) Model**:
    - The final evaluation step involves the Anti-Money Laundering (AML) Model.
    - The AML model analyzes the transaction data to identify suspicious patterns indicative of money laundering.
    - If the AML model indicates a high risk of money laundering, relevant personnel are notified for further investigation.

By following this comprehensive process, the system ensures that financial transactions are thoroughly evaluated for potential fraud and money laundering, enhancing the overall security and reliability of financial operations.

3.5 Tools and Technologies

1. **Python**: Python serves as the foundational programming language for the project, offering simplicity, readability, and versatility. Its extensive standard library and rich ecosystem of third-party packages provide ample support for backend development, data processing, and integration with other technologies.

2. **JavaScript**: JavaScript is instrumental in enabling dynamic, interactive behavior on the project's frontend. As a versatile scripting language, JavaScript facilitates client-side functionalities such as form validation, DOM manipulation, and asynchronous requests, enhancing user engagement and interactivity.

3. **Tkinter**: Tkinter is the most commonly used library for developing GUI (Graphical User Interface) in Python. It is a standard Python interface to the Tk GUI toolkit shipped with Python. As Tk and Tkinter are available on most of the Unix platforms as well as on the Windows system, developing GUI applications with Tkinter becomes the fastest and easiest.

4. **Seaborn**: Seaborn is a visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top

of matplotlib library and also closely integrated to the data structures from pandas. Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs, so that we can switch between different visual representations for same variables for better understanding of dataset. Plots are basically used for visualizing the relationship between variables.

5. **Tensorflow**: TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

6. **Numpy**: Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.

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

4. IMPLEMENTATION DETAILS

4.1 System Architecture

The system architecture of our application is designed with five key components: the Database, Cache Server, Web Backend, Model Backend, and Web Frontend. The Database acts as the central repository, storing all persistent data, including user information and transaction records. The Cache Server is utilized for caching to enhance performance and holds the Celery queue for efficient background task management. The Web Backend handles the core application logic, processing requests from the frontend, interacting with the database for data retrieval and storage, managing background tasks through the cache server, and communicating with the Model Backend for data preprocessing and AI/ML model execution.

The Model Backend is responsible for running AI/ML models, such as the AML Model and the Fraud Detection Model, and providing results to the Web Backend. The Web Frontend is the user interface that displays live information, insights, and transaction reports to the users, and it communicates exclusively with the Web Backend. This streamlined communication flow ensures efficiency, with the frontend sending all requests to the Web Backend, which acts as the central hub managing interactions with the database, cache server, and model backend.

When a user initiates a transaction, the system first checks the type of transaction, categorizing it as a credit card transaction, e-commerce transaction, or bank transaction. The transaction data is then streamed into the system for analysis. The process starts with a reputation list check against previously blacklisted users. If a match is found, the transaction is canceled; otherwise, it moves to the next step. The transaction then undergoes evaluation by the Rules Engine, which applies different rules based on the transaction type and calculates weighted rules. If the transaction fails to meet the defined threshold, it is canceled; if it passes, it is processed by the appropriate fraud detection AI model. The AI model, specific to the transaction type, analyzes the data for fraudulent activities. If fraud is detected, the transaction is canceled; if not, it proceeds to the AML Model for final evaluation. The AML Model assesses the transaction for money laundering risks, and if a high risk is detected, relevant personnel are notified for further investigation. This comprehensive process ensures thorough evaluation of transactions for fraud and money laundering, enhancing the system's security and reliability.

4.2 Implemented model

4.2.1 Dataset

To train and validate our models, we sourced diverse and comprehensive datasets from Kaggle, which include various types of online financial transactions. These datasets are essential for developing robust fraud detection models, providing a wide range of scenarios and data points that enhance the system's learning and detection capabilities[8].

1. **Credit Card Fraud Datasets**

   The dataset utilized in this project contains transactions made by credit cards in September 2013 by European cardholders. It encompasses transactions over a span of two days, totaling 284,807 transactions, out of which 492 are fraudulent. This results in a highly imbalanced dataset where the positive class (frauds) accounts for only 0.172% of all transactions[7]. The graph below shows the data disrtibution between fraudulent and Non-Fraudulent data present in the dataset.

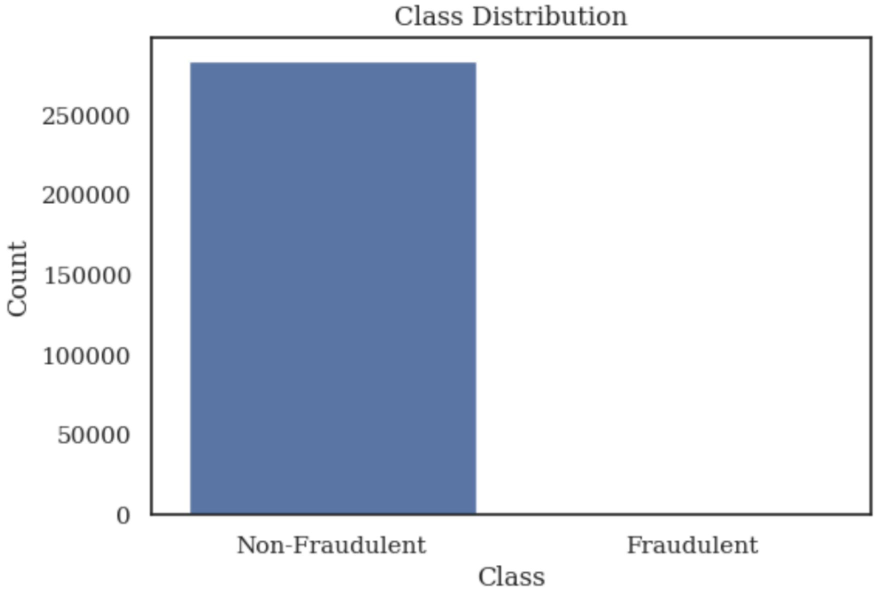


Figure 5: Graph showing Data Distribution

Dataset contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are ’Time’ and ’Amount’. Feature ’Time’ contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature ’Amount’ is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. Feature ’Class’ is the response variable and it takes value 1 in case of fraud and 0 otherwise.

2. **E-commerce Fraud Datasets**

   For this project, we utilized a comprehensive dataset extracted from Kaggle, encompassing approximately 150,000 e-commerce transactions[13]. The features include sign up time, purchase time, purchase value, device id, user id, browser, and IP address. We added a new feature that measured the time difference between sign up and purchase, as the age of an account is often an important variable in fraud detection. The graph below shows dataset that has number of transaction passed and transaction failed

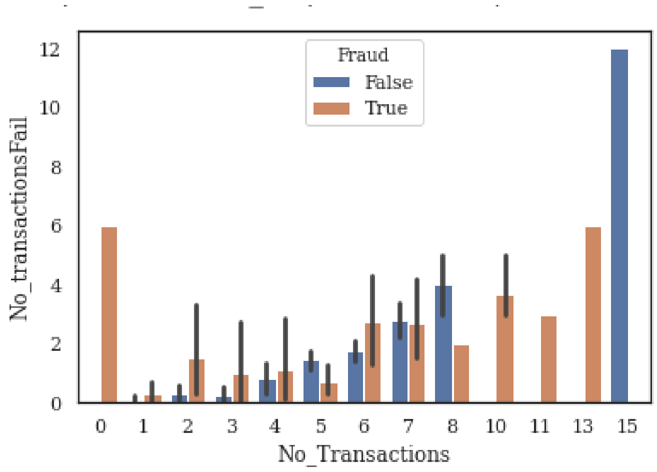


Figure 6: Graph of Number of transactions vs number of transactionss failed

3. **Bank transaction Datasets** This synthetic tabular dataset comprises 1M instances. Each instance represents a credit card application, for which the values of 31 features and the corresponding label is available. The label indicates whether the application is fradulant. The features capture various information associated with the applicant (such as the their age, income, housing status, employment and credit risk score) or the application (such as the proposed credit limit

and the month of the application). The dataset contains a combination of numerical (continuous and discrete) and categorical features and some of the features have missing values[9] .

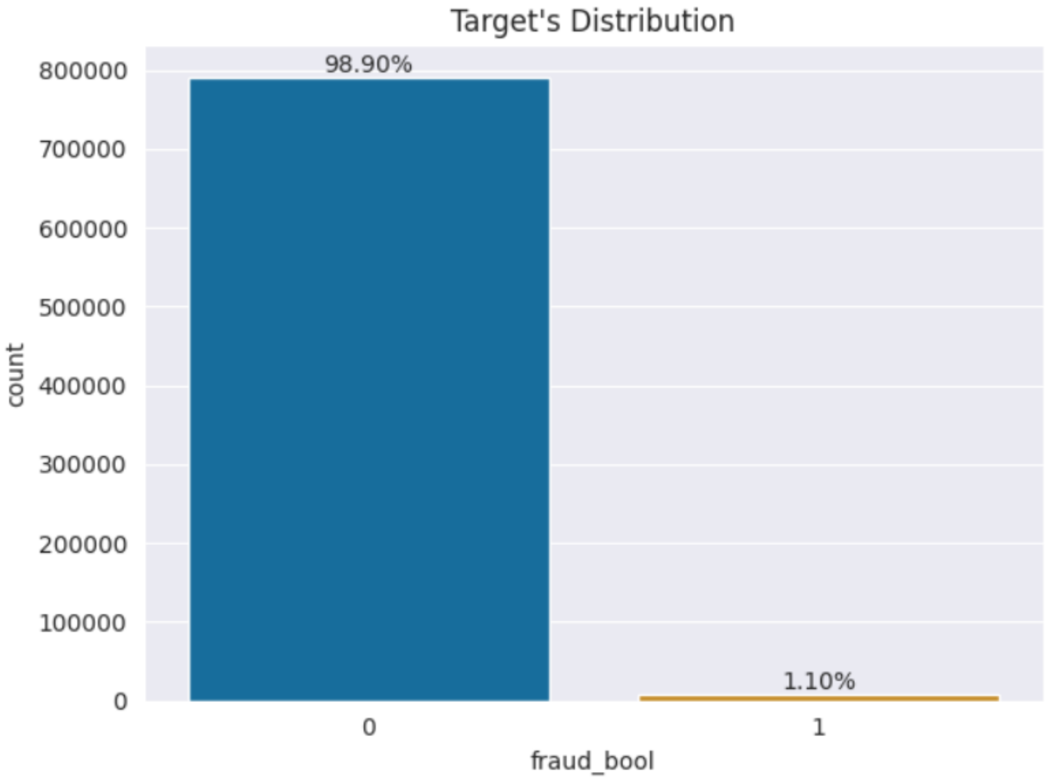


Figure 7: Graph of Target Distribution of data

Quoting authors' claims, the BAF datasets suite, comprising the main dataset considered in this project and its variations[9], are

(a) Realistic, based on a present-day real-world dataset for fraud detection;

(b) Biased, each dataset has distinct controlled types of bias;

(c) Imbalanced, this setting presents a extremely low prevalence of positive class;

(d) Dynamic, with temporal data and observed distribution shifts; Privacy preserving, to protect the identity of potential applicants we have applied differential privacy techniques (noise addition), feature encoding and trained a generative model (CTGAN).

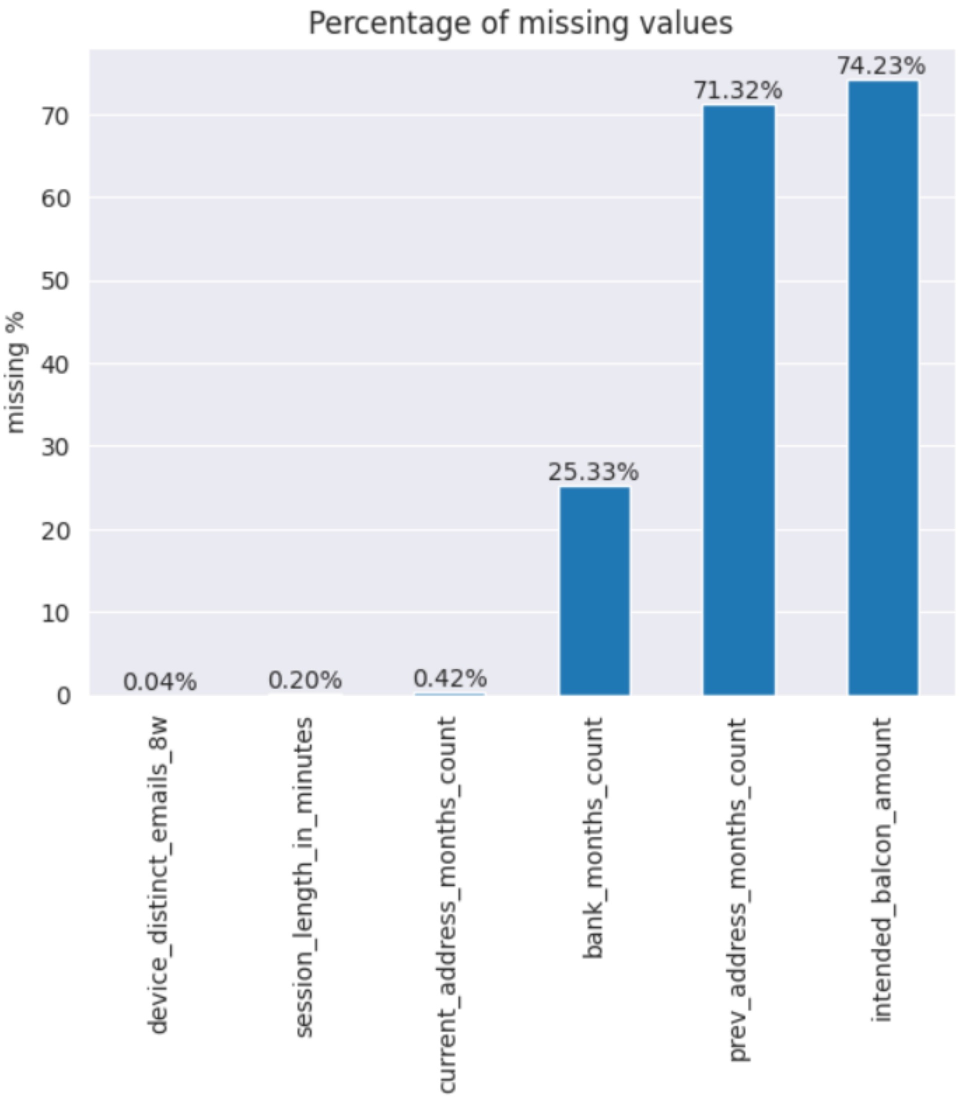


Figure 8: Graph of data features

4. **Anti Money Laundering Transaction Data (SAML-D)**

Money laundering remains a significant global issue, driving the need for improved transaction monitoring methods. Current anti-money laundering (AML) procedures are inefficient, and access to data is difficult/restricted by legal and privacy issues. Moreover, existing data often lacks diversity and true labels. This study introduces a novel AML transaction generator, creating the SAML-D dataset with enhanced features and typologies, aiming to aid researchers in evaluating their models and developing more advanced monitoring methods [12].

The dataset incorporates 12 features and 28 typologies (split between 11 normal and 17 suspicious). These were selected based on existing datasets, the academic literature, and interviews with AML specialists. The dataset comprises 9,504,852 transactions, of which 0.1039% are suspicious. It also includes 15 graphical network structures to represent the transaction flow within these typologies. The structures, while sometimes shared among typologies, vary significantly in parameters to increase complexities and challenge detection efforts[12].

- Time and Date: Essential for tracking transaction chronology.
- Sender and Receiver Account Details: Helps uncover behavioural patterns and complex banking connections.
- Amount: Indicates transaction values to identify suspicious activities.
- Payment Type: Includes various methods like credit card, debit card, cash, ACH transfers, cross-border, and cheque.
- Sender and Receiver Bank Location: Pinpoints high-risk regions including Mexico, Turkey, Morocco, and the UAE.
- Payment and Receiver Currency: Align with location features, adding complexity when mismatched.

- 'Is Suspicious' Feature: Binary indicator differentiating normal from suspicious transactions.
- Type: Classifies typologies, offering deeper insights.

4.2.2 Data Preprocessing

In our project, we employ several preprocessing techniques to prepare the datasets for effective model training. These techniques ensure that the data is in the appropriate format and scale, enhancing the model's ability to detect fraud. Here are the key preprocessing steps:

Missing Flagger:

The MissingFlagger class creates a binary 'missing flag' feature for some of the features with missing values. The values 0 and 1 indicate non-missing and missing values, respectively. This transformation helps the model learn patterns associated with missing values, which can be informative for predicting fraud.

Categorical Converter:

The CategoricalConverter class converts the dtype of categorical features to 'category'. This is particularly useful for classifiers that can natively handle categorical features, such as the LightGBM classifier. By converting these features, the model can better utilize the categorical information.

OneHotEncoder:

One-hot encoding is a technique used to convert categorical variables into a binary (0 or 1) format that machine learning models can process. The CustomOneHotEncoder class handles this transformation, ensuring that categorical variables are appropriately represented for the model.

Scaler:

The Scaler standardizes the specified columns (or all columns if none are specified), transforming them to have a mean of 0 and a standard deviation of 1. Standardization is crucial for many machine learning algorithms, as it ensures that all features contribute equally to the model's training process.

These preprocessing steps collectively enhance the quality and consistency of the data, making it more suitable for training robust fraud detection models.

4.2.3 Model

1. **LightGBM (Light Gradient Boosting Machine for Bank Fraud Detection):**

   LightGBM is part of the boosting family of algorithms. Boosting involves training multiple weak learners (typically decision trees) sequentially, where each subsequent model aims to correct the errors of the previous ones. LightGBM enhances this approach through several optimizations:

- **Gradient-based One-Side Sampling (GOSS)**:

  Focuses on data points with larger errors, speeding up training without sacrificing accuracy.

- **Exclusive Feature Bundling**:

  Combines mutually exclusive features, reducing feature count and memory usage.

- **Histogram-based Decision Tree Learning**:

  Uses histograms to bucket continuous features, enhancing computation speed.

- **Leaf-wise Tree Growth**:

  Grows trees leaf-wise, reducing loss more effectively, suitable for complex datasets.

Justification for using LightGBM :

- **High-dimensional Data**: Efficient in handling numerous features related to customer behavior and transaction details.
- **Imbalanced Data**: Effectively manages rare fraud cases and improves detection rates.
- **Complex Patterns**: Captures intricate interactions between features through its tree-based approach.
- **Speed and Efficiency**: Allows frequent model training and updates to adapt to new fraud patterns.
- **Handling of Categorical Features**: Natively processes categorical data common in banking, such as account and transaction types.

2. **XGBoost (Extreme Gradient Boosting)**

   XGBoost (Extreme Gradient Boosting) is a highly efficient and scalable machine learning algorithm that is widely used for classification and regression tasks. It is a type of gradient boosting algorithm that builds an ensemble of decision trees in a sequential manner, where each tree aims to correct the errors of its predecessor.

   Justification for using XGBoost in Anti Money Laundering

   (a) **High Accuracy and Performance:** XGBoost excels at handling complex and high-dimensional data, which is crucial for accurately identifying subtle patterns of money laundering. Its advanced boosting techniques improve predictive performance by combining multiple weak learners into a robust model, resulting in higher accuracy and reduced false positives in detecting suspicious activities.

   (b) **Effective Handling of Imbalanced Data:**

   AML datasets often have a significant imbalance between fraudulent and non-fraudulent transactions. XGBoost's ability to handle imbalanced data through techniques like weighted class distribution and tailored objective functions helps in accurately identifying rare fraudulent cases without being overwhelmed by the majority class of legitimate transactions.

   (c) **Feature Importance and Interpretability:**

   XGBoost provides insights into the importance of different features, which helps in understanding which attributes are most indicative of money laundering. This interpretability aids in refining models, ensuring regulatory compliance, and focusing on key indicators for improved detection and prevention strategies.

3. **Logistic Regression**

   Logistic regression is a statistical method used for binary classification problems. It predicts the probability that a given input belongs to a certain class. The output of logistic regression is a value between 0 and 1, which represents the probability of the dependent variable being in a particular class. To Train the E-commerce dataset and Card Fraud Detection Logistic regression has been used.

   Justification for using Logistic Regression

   (a) Logistic regression can help predict the Fraud column based on the other features, such as transactionAmount, paymentMethodType, No_Transactions, No_Orders, No_Payments, and other customer-related details.

   (b) It is specifically designed for binary outcomes, which is the case with the Fraud variable (True or False).

   (c) Logistic regression can handle both continuous and categorical features. Your dataset includes both types, such as transactionAmount (continuous) and paymentMethodType (categorical).

5. RESULT AND ANALYSIS

5.1 Money Laundering Prediction

For predicting Money Laundering chances we have used the preprocessed model was used. The confusion matrix of Money Laundering detection model is shown below:

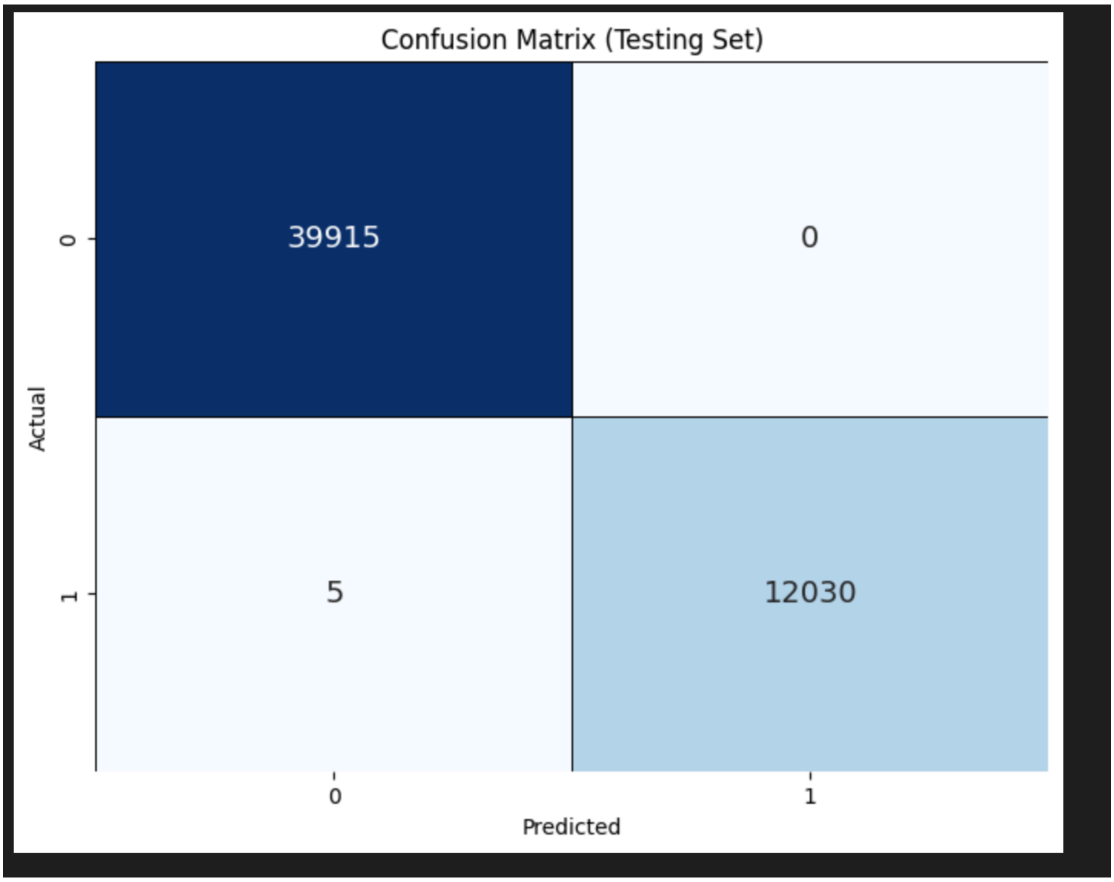


Figure 9: Confusion Matrix for Money laundering Detection

5.2 Fraud Detection

The fraud detection system was rigorously evaluated across three types of transactions: credit card, e-commerce, and bank transactions. For credit card transactions, the system successfully identified unauthorized charges and suspicious spending patterns, minimizing false positives. In the e-commerce domain, it effectively detected unusual purchasing behaviors and fraudulent account activities. The evaluation of bank transactions demonstrated the system's capability to recognize irregular transaction patterns and potential vulnerabilities. These comprehensive evaluations across different transaction types showcase the system's robust performance and adaptability in identifying fraudulent activities in various financial contexts.

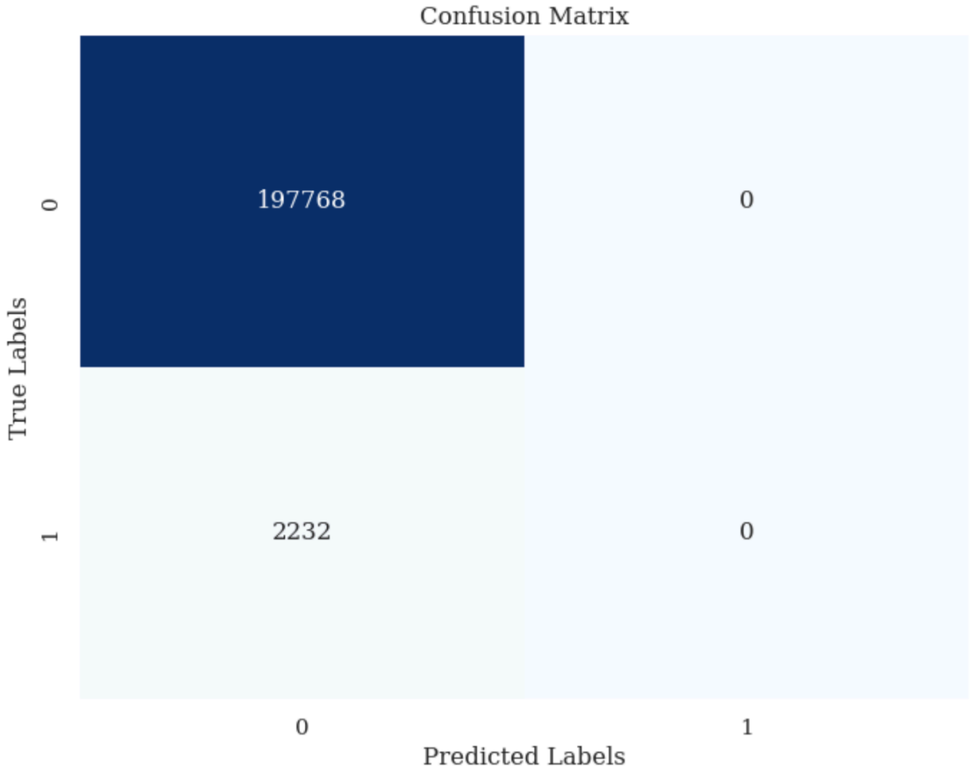


Figure 10: Confusion Matrix for Bank Transaction Fraud Detection

In the figure above presents the confusion matrix for bank transaction fraud detection. The matrix shows the performance of the fraud detection model by displaying the number of true labels and predicted lables.

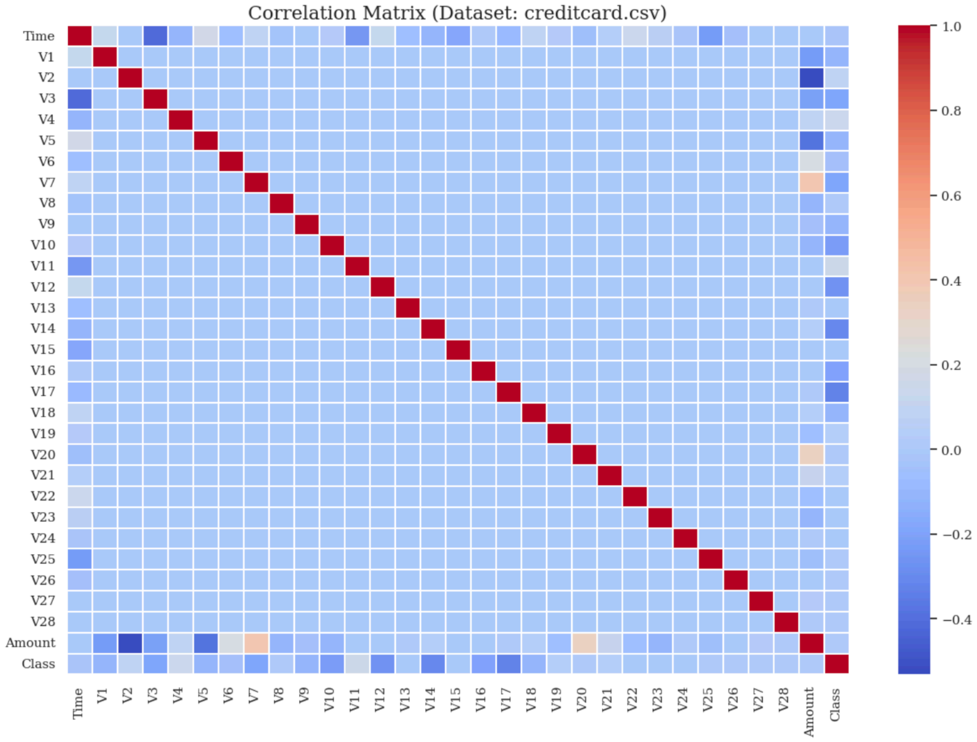
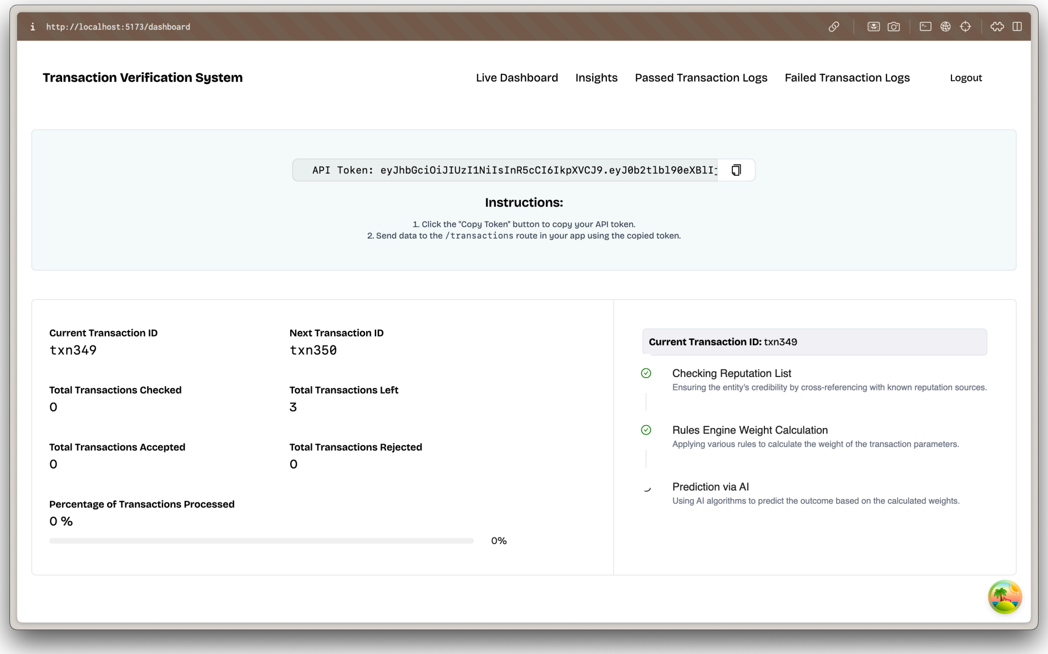


Figure 11: corelation matrix of crefit card fraud detection

5.3 Insights

In this section, we present the results and analysis of the transaction processing system, focusing on the insights derived from the live dashboard interface and detailed transaction evaluations.



Transaction Verification System

API Token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlI

Instructions:

1. Click the "Copy Token" button to copy your API token.
2. Send data to the `/transactions` route in your app using the copied token.

Current Transaction ID txn349	**Next Transaction ID** txn350
Total Transactions Checked 0	**Total Transactions Left** 3
Total Transactions Accepted 0	**Total Transactions Rejected** 0
Percentage of Transactions Processed 0 %	

0%

Current Transaction ID: txn349

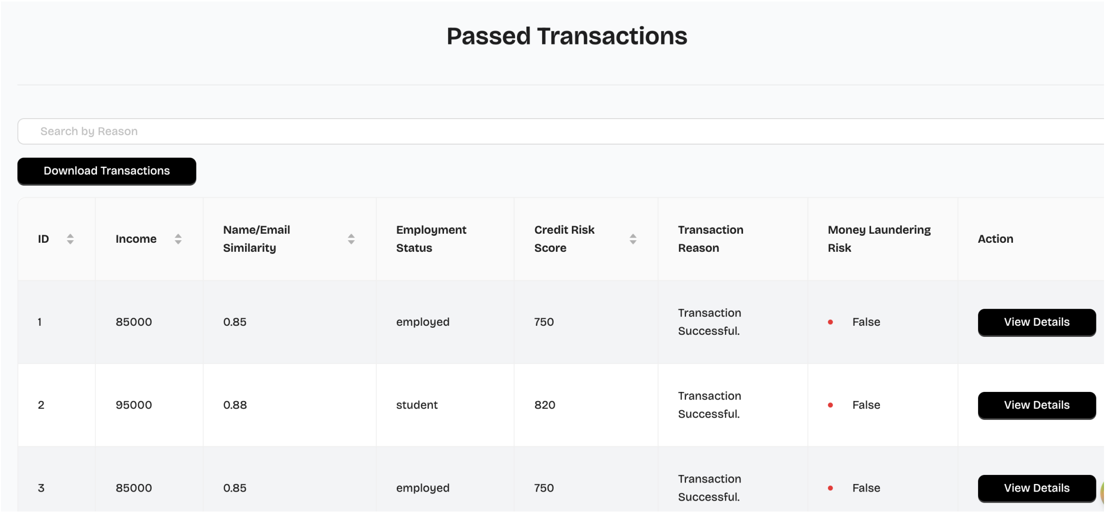
✓ Checking Reputation List
Ensuring the entity's credibility by cross-referencing with known reputation sources.

✓ Rules Engine Weight Calculation
Applying various rules to calculate the weight of the transaction parameters.

Prediction via AI
Using AI algorithms to predict the outcome based on the calculated weights.

Figure 12: Interface of the live dashboard

Each transaction data enters the system and is dynamically displayed on the live dashboard. The transactions are queued and processed sequentially through several stages, as depicted in the User Interface as shown in figure 11. These stages include Reputational List Analysis, Rules Engine, Fraud Detection AI Models, and Money Laundering Detection AI Models. After the completion of the evaluation of all the transactions the summary is shown depicting the number of passed and failed transactions.



Passed Transactions

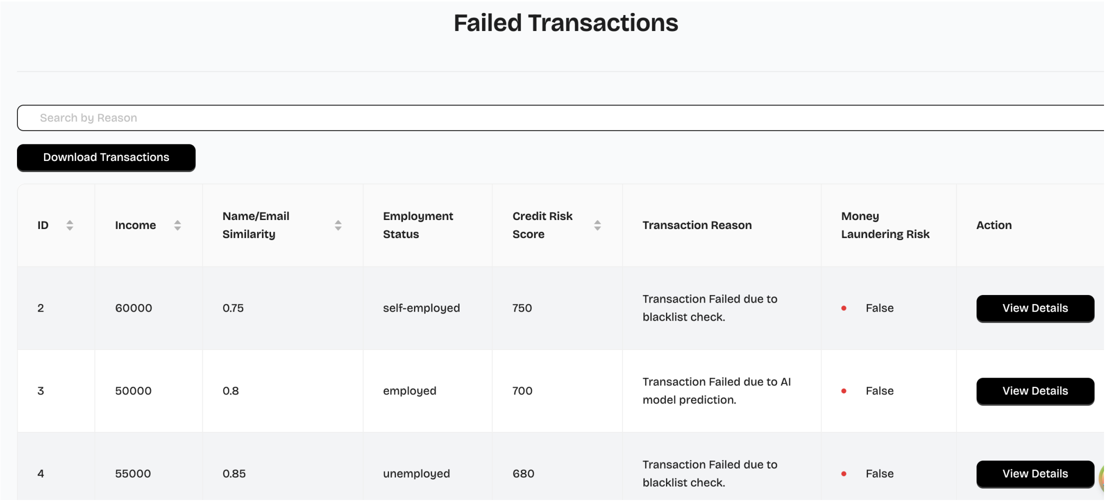
Search by Reason

Download Transactions

ID	Income	Name/Email Similarity	Employment Status	Credit Risk Score	Transaction Reason	Money Laundering Risk	Action
1	85000	0.85	employed	750	Transaction Successful.	● False	View Details
2	95000	0.88	student	820	Transaction Successful.	● False	View Details
3	85000	0.85	employed	750	Transaction Successful.	● False	View Details

Figure 13: Interface of Passed Transaction

In Figure 12 above, the completion of a transaction is illustrated, showcasing transactions that have successfully passed all necessary steps. The processed and successful transaction data is presented in a well-organized table format. This table offers a clear and comprehensive view of the completed transactions, detailing each transaction's attributes and relevant metrics. The structured presentation allows users to easily access and interpret the data, providing valuable insights into transaction performance and outcomes. By displaying successful transactions in this format, users can efficiently review and analyze the data, track performance trends, and gain a deeper understanding of transaction success factors. This clarity in presentation supports better decision-making and facilitates effective monitoring of transaction processing.



Failed Transactions

ID	Income	Name/Email Similarity	Employment Status	Credit Risk Score	Transaction Reason	Money Laundering Risk	Action
2	60000	0.75	self-employed	750	Transaction Failed due to blacklist check.	● False	View Details
3	50000	0.8	employed	700	Transaction Failed due to AI model prediction.	● False	View Details
4	55000	0.85	unemployed	680	Transaction Failed due to blacklist check.	● False	View Details

Figure 14: Interface of Failed Transaction

Similarly, in Figure 14, transactions that fail to meet the required criteria at various stages are depicted in detail. This figure illustrates the failed transactions by presenting them in a comprehensive table format. The table is meticulously organized to provide a clear and structured view of each failed transaction, including specific details about the reasons for failure. This format allows users to easily identify and analyze the underlying causes behind each failed transaction, such as discrepancies in data, rule violations, or system errors. By presenting this information in a clear and accessible manner, users can gain valuable insights into the failure patterns, facilitating targeted troubleshooting and process improvements. The table not only enhances transparency but also supports more effective decision-making by highlighting critical areas that require attention or corrective action.

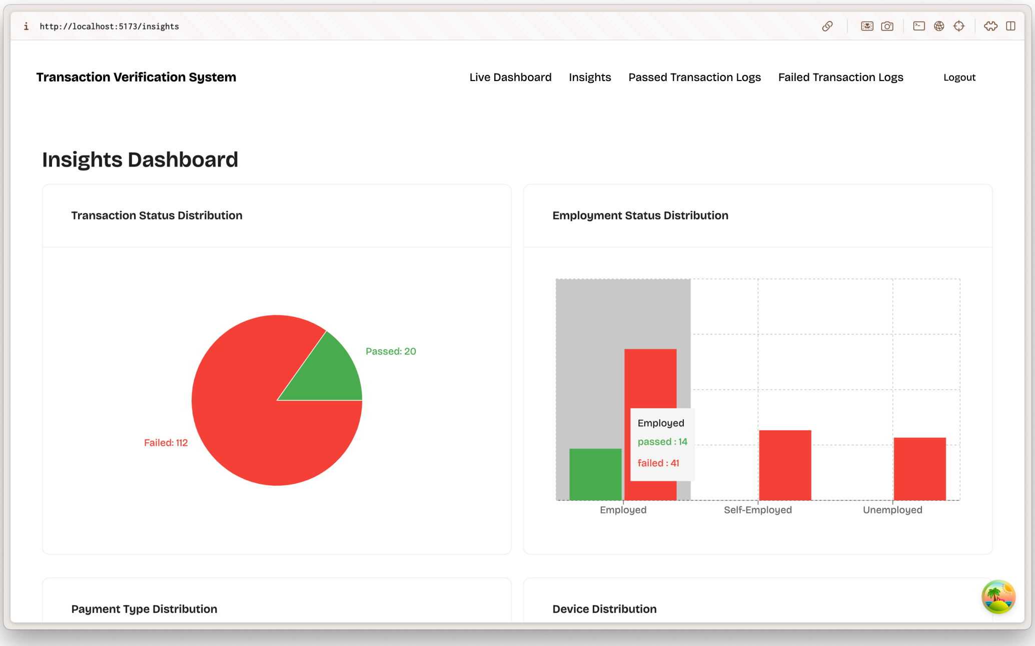


Figure 15: Graphical representation and insights of the data

The user is provided with a comprehensive and interactive visual representation of data transactions through various graphs and charts. These visual insights highlight different attributes of the data, such as failed and passed transactions, and other critical metrics. The graphs and charts are designed to present trends, patterns, and key attributes of the data clearly, allowing users to gain an intuitive understanding of complex information. By displaying real-time and historical data, these visual tools enable users to quickly identify significant anomalies, track transaction performance over time, and analyze key attributes related to the success and failure of transactions. This enhanced visibility facilitates informed decision-making and effective responses to critical information as shown in figure 15.

5.4 Performance Analysis:

Our system demonstrates notable advancements in banking fraud and money laundering detection, as illustrated by the following key performance aspects:

- **Enhanced Fraud Detection Accuracy:** The specialized fraud detection model is tailored to unique banking transaction features, significantly improving the accuracy of identifying fraudulent activities. This model effectively reduces false positives and negatives, ensuring smoother processing of legitimate transactions and minimizing disruptions.
- **Effective AML Integration:** Advanced AML algorithms assess the likelihood of transactions being related to money laundering. By calculating a percentage chance for each transaction and flagging those that exceed a specific threshold, the system enables early detection of potential money laundering activities, facilitating timely intervention.
- **Real-Time Processing and Communication:** The integration of WebSocket technology ensures real-time updates and communication. This capability provides users and banking staff with immediate access to transaction information, enhancing responsiveness and user engagement compared to traditional periodic update systems.
- **User-Friendly Interface:** The system features a modern, intuitive interface designed to enhance user experience. The streamlined design facilitates efficient navigation and quick access to critical in-

formation, reducing the learning curve and improving overall user satisfaction.

- **Comprehensive Analytics and Insights:** The system offers detailed analytics and visualizations of transaction data, including trends and anomalies. This functionality supports informed decision-making by providing actionable insights into transaction performance and potential risks.

- **Multistep Detection Process:** The multi-step approach integrates initial screening, fraud detection models, rule-based calculations, reputation checks, and AML risk assessments. This thorough process ensures a robust evaluation of each transaction, improving detection accuracy and overall security.

6. PROJECT TASK AND TIME SCHEDULE

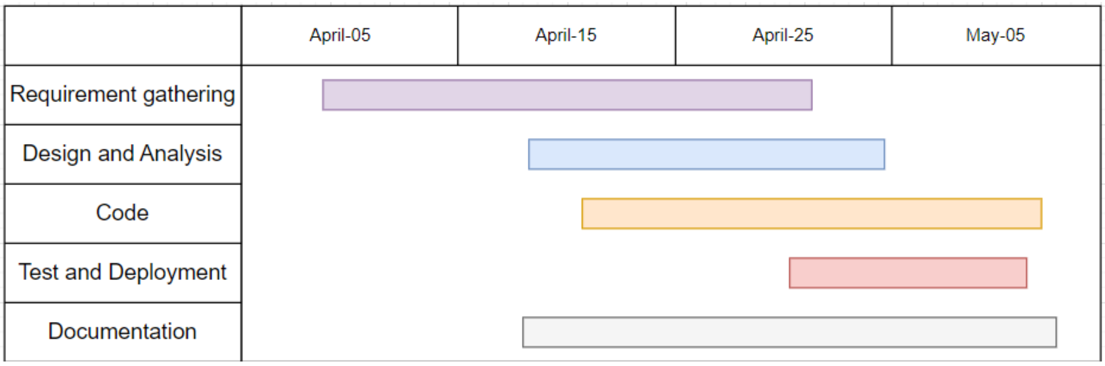


Figure 16: Gantt chart for first Increment

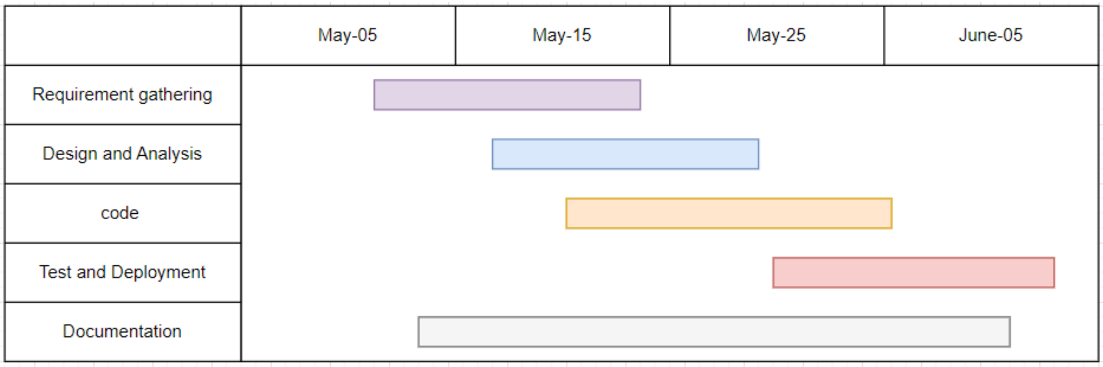


Figure 17: Gantt chart for second Increment

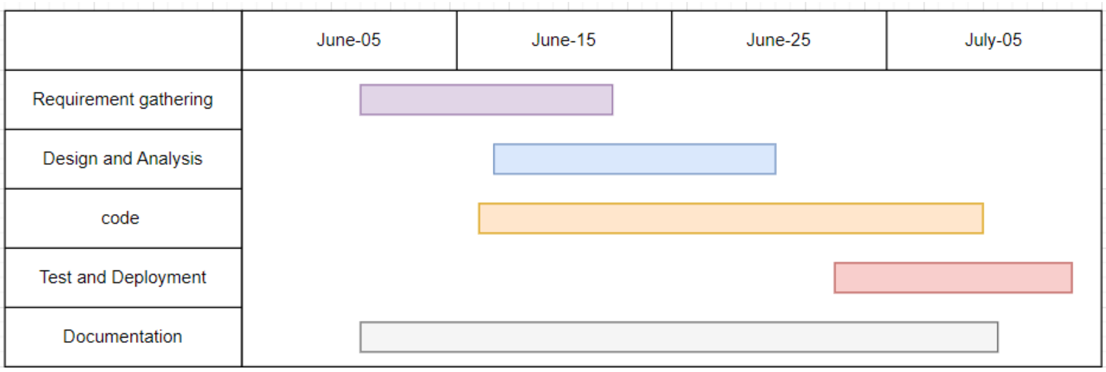


Figure 18: Gantt chart for third Increment

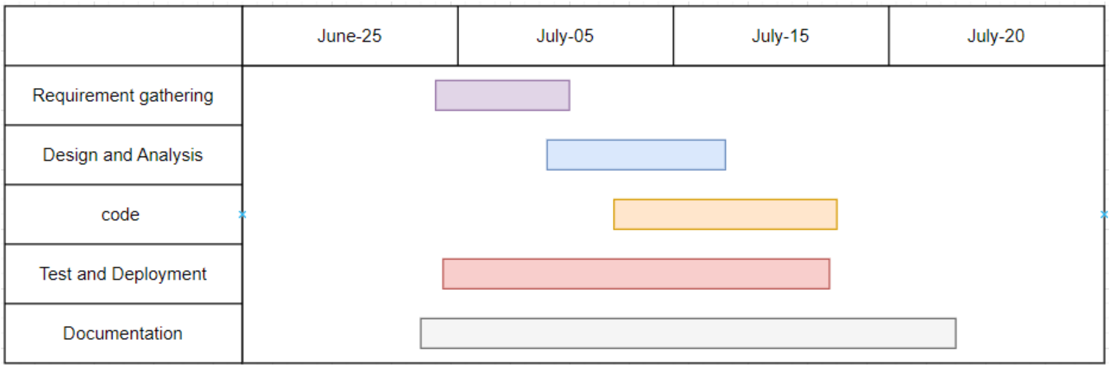


Figure 19: Gantt chart for final Increment

7. CONCLUSION

In conclusion, the Transaction Integration Verification System (TIVS) is a state-of-the-art solution designed to address the pressing security and integrity challenges of online financial transactions. TIVS harnesses advanced artificial intelligence (AI) algorithms to provide real-time monitoring and analysis, ensuring the detection and prevention of fraudulent activities. By integrating anomaly detection, pattern recognition, and behavior analysis, TIVS can continuously scrutinize transactional data, promptly identifying irregular trends and suspicious activities. When potential threats are detected, TIVS alerts financial institutions and regulatory bodies, facilitating swift and effective responses to mitigate risks.

Our project demonstrates a sophisticated approach to detecting banking fraud and money laundering through tailored models, real-time communication, and advanced analytics. We employ three different models and three datasets specifically for credit card fraud detection, E-commerce fraud detection, and bank transaction fraud detection. The multistep process, combined with effective AML measures and modern design principles, sets a new standard in the industry. The system's comprehensive approach addresses the complexities of financial transactions, providing robust protection and a superior user experience.

By offering a robust and scalable solution for real-time fraud detection and prevention, TIVS ensures that financial transactions remain secure, thereby maintaining public trust in the financial ecosystem. TIVS aims to safeguard the financial system by detecting and preventing financial frauds in real-time, and by providing timely alerts to relevant authorities, enabling proactive measures against these threats. Through its real-time capabilities and advanced analytical methods, TIVS plays a vital role in upholding the integrity of online financial systems and contributing to a more secure and trustworthy financial landscape.

8. FUTURE WORK

Future work on this project will focus on the following areas:

1. Explore advanced and hybrid algorithms to enhance detection accuracy and efficiency.
2. Incorporate additional data types to provide a more comprehensive view of transactions and improve fraud detection.
3. Refine real-time adaptation and anomaly detection to maintain high detection rates.
4. Expand reporting features and integrate automated compliance checks to meet evolving regulatory requirements.
5. Enhance user experience and interface with intuitive dashboards and customizable features.
6. Optimize system performance and scalability to handle increasing transaction volumes.

REFERENCES

[1] Alex Beutel, Leman Akoglu, and Christos Faloutsos. “Graph-based user behavior modeling: from prediction to fraud detection”. In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. ACM. 2015, pp. 2309–2310.

[2] Siddhartha Bhattacharyya et al. “Data mining for credit card fraud: A comparative study”. In: *Decision Support Systems* 50.3 (2011), pp. 602–613.

[3] Tianqi Chen and Carlos Guestrin. “XGBoost: A scalable tree boosting system”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2016, pp. 785–794.

[4] DataVisor Inc. *DataVisor API References*. Accessed: 2024-07-22. 2024. URL: `https://www.datavisor.com/datavisor-api-guide/`.

[5] Habib Dhieb et al. “Fraud detection in the insurance sector using machine learning”. In: *Journal of Financial Technology* 10.2 (2020), pp. 123–136.

[6] FICO. *FICO Fraud Detection and AML Solutions*. Accessed: 2024-07-26. 2024. URL: `https://www.fico.com/en/products/fraud-detection/`.

[7] Machine Learning Group. *Credit Card Fraud Detection*. 2018. URL: `https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud`.

[8] Prince Grover et al. *Fraud Dataset Benchmark and Applications*. 2023. arXiv: `2208.14417 [cs.LG]`.

[9] Sérgio Jesus. *Bank Account Fraud Dataset Suite (NeurIPS 2022)*. 2022. URL: `https://www.kaggle.com/datasets/sgpjesus/bank-account-fraud-dataset-neurips-2022?select=Base.csv`.

[10] NICE Actimize. *NICE Actimize Financial Crime, Risk, and Compliance Solutions*. Accessed: 2024-07-26. 2024. URL: `https://www.nice.com/financial-crime-risk-compliance/`.

[11] Guanglei Niu et al. “A novel boosting model for anti-money laundering”. In: *IEEE Access* 8 (2020), pp. 7600–7610.

[12] B. Oztas et al. “Enhancing Anti-Money Laundering: Development of a Synthetic Transaction Monitoring Dataset”. In: *2023 IEEE International Conference on e-Business Engineering (ICEBE)*. Sydney, Australia, 2023, pp. 47–54. DOI: 10.1109/ICEBE59045.2023.00028.

[13] Binh Vu. *Fraud ecommerce*. 2022. URL: https://www.kaggle.com/datasets/vbinh002/fraud-ecommerce.