VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

FINAL PROJECT REPORT

CREDIT CARD FRAUD DETECTION SYSTEM

SOFTWARE ENGINEERING BCSE301L

MISBAH ANWAR 21BAI1439   NEHA RAMESH 21BAI1452   FAZEEL KHURSHID 21BAI1820

CONTENTS

INTRODUCTION

Credit card fraud detection is crucial for financial security, aiming to prevent unauthorized transactions. With the rise of electronic transactions, advanced detection technologies are needed to combat increasing fraud attempts. It aims to prevent financial loss by analysing transaction data in real-time to detect anomalies in spending patterns. Modern systems use machine learning, rule-based systems, and anomaly detection methods to identify potential fraud early.

In recent years, the proliferation of electronic transactions has led to a corresponding increase in credit card fraud. Fraudsters have become increasingly sophisticated, employing various techniques to bypass traditional fraud detection methods. To address this growing threat, financial institutions and credit card companies have implemented advanced fraud detection systems. These systems utilize a combination of machine learning algorithms, rule-based systems, and anomaly detection techniques to analyze transaction data in real-time.

By monitoring spending patterns, transaction frequency, geographic location, and other relevant factors, these systems can identify suspicious activities and potential fraud attempts as they occur. Machine learning algorithms play a crucial role in this process by continuously learning from past transaction data to improve their ability to detect fraudulent behavior.

Rule-based systems complement these algorithms by allowing for the customization of specific rules and thresholds to flag potentially fraudulent transactions.

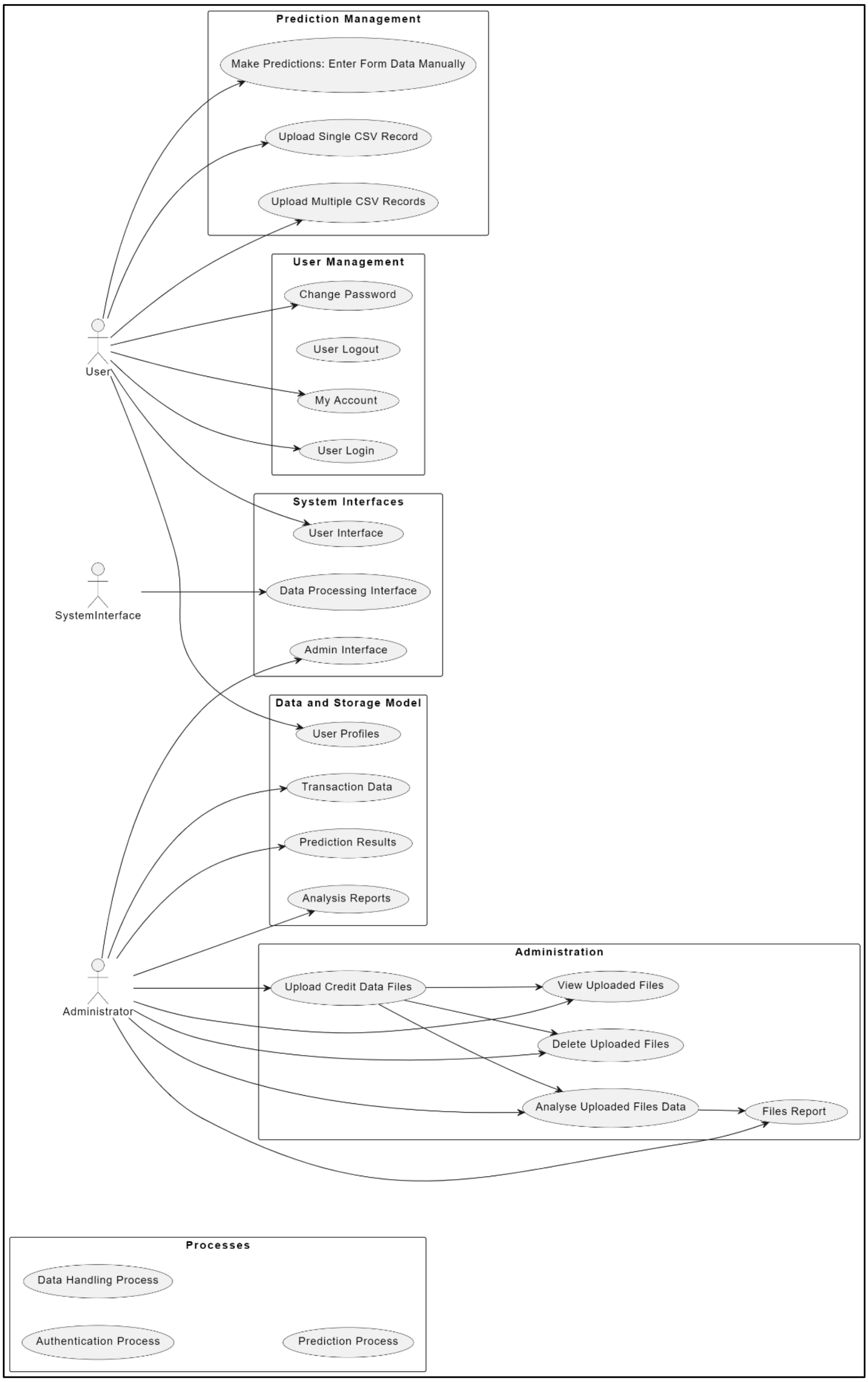
Anomaly detection techniques further enhance the effectiveness of fraud detection systems by identifying unusual or unexpected patterns in transaction data.

By comparing each transaction to established norms and historical data, these techniques can quickly identify deviations that may indicate fraudulent activity.

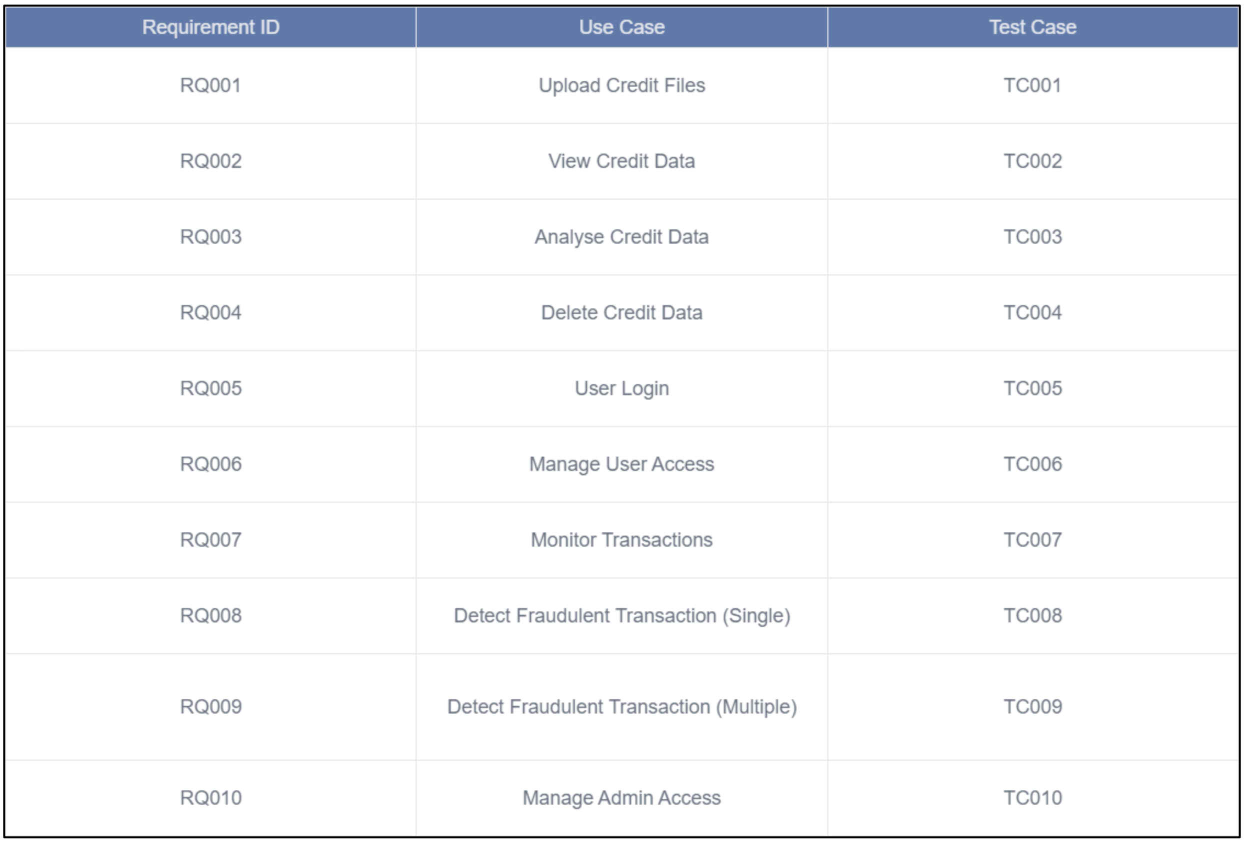
Overall, the combination of these advanced technologies enables financial institutions and credit card companies to detect and prevent credit card fraud in real-time, helping to protect consumers and businesses from financial loss and preserve the integrity of the financial system.

REQUIREMENTS

USE CASE DIAGRAM



TRACEABILITY MATRIX



Requirement ID	Use Case	Test Case
RQ001	Upload Credit Files	TC001
RQ002	View Credit Data	TC002
RQ003	Analyse Credit Data	TC003
RQ004	Delete Credit Data	TC004
RQ005	User Login	TC005
RQ006	Manage User Access	TC006
RQ007	Monitor Transactions	TC007
RQ008	Detect Fraudulent Transaction (Single)	TC008
RQ009	Detect Fraudulent Transaction (Multiple)	TC009
RQ010	Manage Admin Access	TC010

FULLY DRESSED DESCRIPTIONS

Use Case 1: Detect Fraudulent Transactions (Multiple) (RQ009)

Actors: Fraud Analyst, Customer.

Preconditions: Customer provides their credit card data due to suspicion.

Basic Flow: The analyst tests the data using the model and reviews the details of the flagged transaction.

The analyst updates the case status and takes action if necessary

Alternative Flow: If the investigation reveals a false positive, the analyst updates the system to improve future accuracy.

Use Case 2: Manage Admin Access (RQ010)

Actors: Admin.

Preconditions: Valid login credentials for the admin. Authorization to manage admin access.

Basic Flow: Login with valid credentials. Navigate to Manage Files. Admin has access to all Files.

Alternate Flow: Invalid Login Credentials- System displays error. Re-enter valid credentials.

DOMAIN MODEL

Core Entities and Functions

1. User Management

User Login: Authenticates users to access their accounts and system functionalities.

User Logout: Securely logs out users from the system.

Change Password: Allows users to update their password.

My Account: Provides users with access to their account details and settings.

2. Administration

Upload Credit Data Files: Enables administrators to upload CSV files containing credit transaction data.

View Uploaded Files: Lists all the credit data files uploaded to the system.

Delete Uploaded Files: Allows deletion of previously uploaded files.

Analyse Uploaded Files Data: Analytical tools to explore and analyse data from uploaded files.

Files Report: Generates reports based on the analysis of uploaded files, summarizing key insights and findings.

3 . Prediction Management

Make Predictions: Enter Form Data Manually: Users can input transaction data through a form to predict if it's fraudulent.

Upload Single CSV Record: Allows for uploading a single record in CSV format for fraud prediction.

Upload Multiple CSV Records: Enables bulk upload of multiple records in CSV format for simultaneous fraud predictions.

4. Data and Storage Model

User Profiles: Store user credentials, roles (e.g., admin, analyst), and personal settings.

Transaction Data: Maintains records of all transaction data uploaded, including details necessary for fraud analysis.

Prediction Results: Stores the outcomes of fraud predictions, linking them to specific transactions or data files.

Analysis Reports: Archives generated reports from analysed transaction data for historical reference and audit purposes.

5. System Interfaces

User Interface: Provides a web-based interface for user interactions with functionalities like login, account management, file uploads, and predictions.

Admin Interface: Specialized dashboard for system administrators for file management, user management, and system settings.

Data Processing Interface: Handles the backend logic for data analysis, predictions, and report generation.

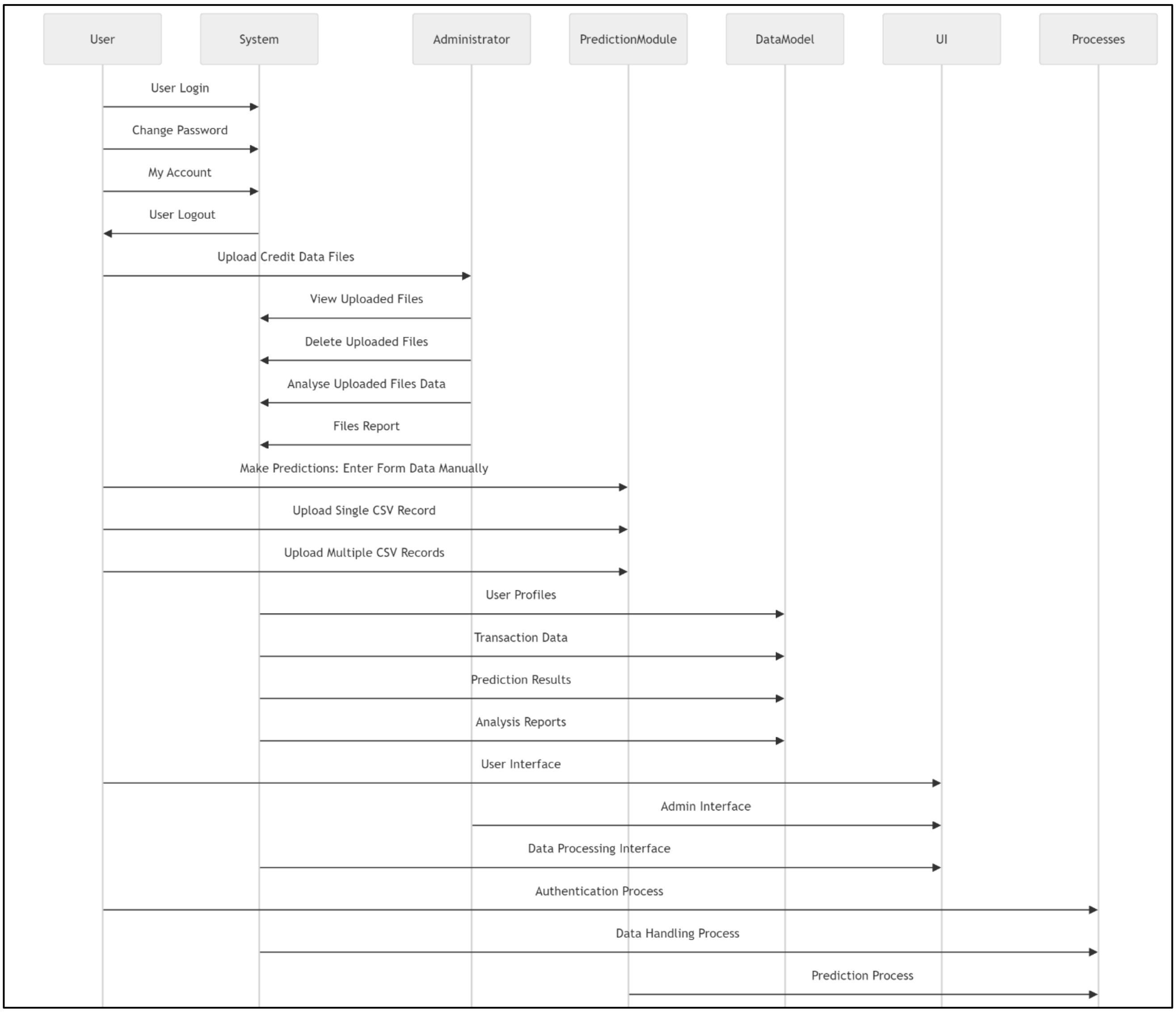
6. Processes

Authentication Process: Manages the login and logout functionalities, ensuring secure access to the system.

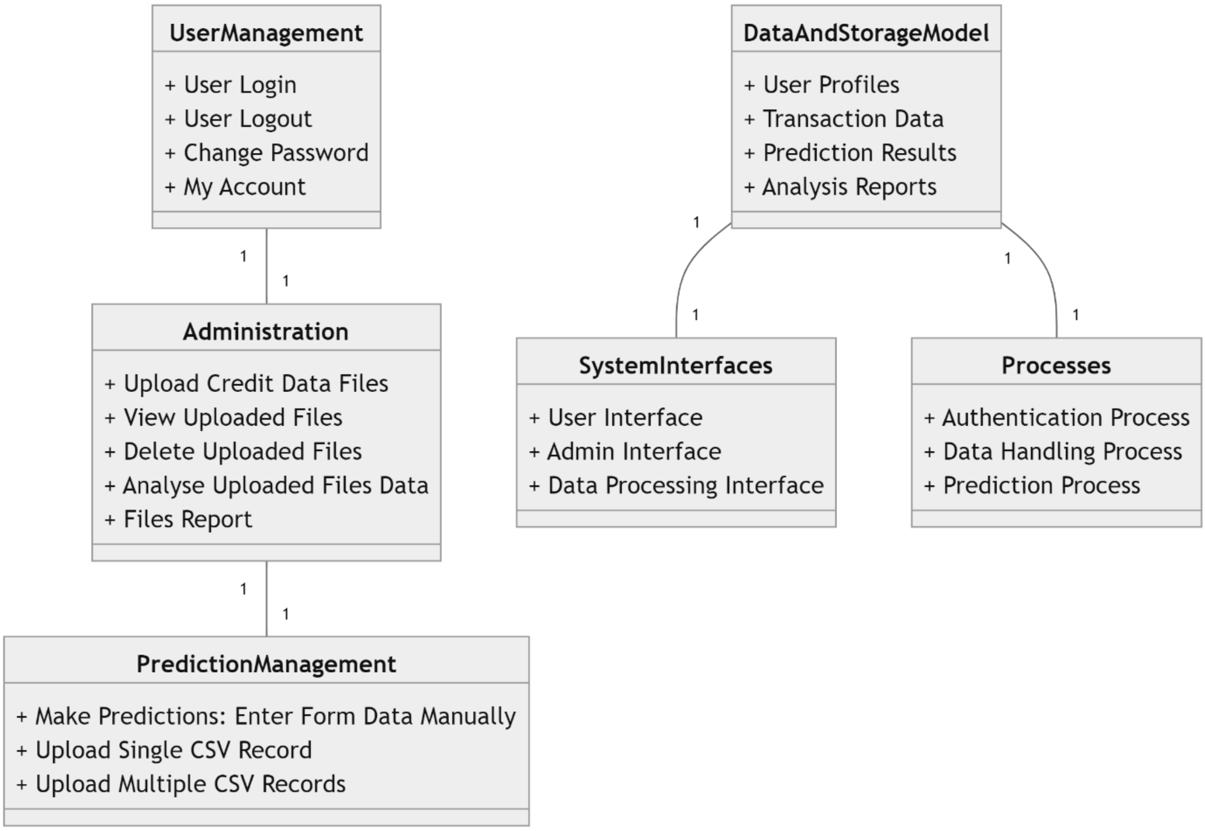
Data Handling Process: Includes uploading, storing, deleting, and analysing transaction data.

Prediction Process: Employs machine learning models and rule-based systems to conduct fraud predictions based on uploaded or manually entered data.

INTERACTION DIAGRAM



1. CLASS DIAGRAM AND INTERFACE SPECIFICATION



1. **User Interface**

   login(username: string, password: string): bool

   Authenticates user with provided username and password.

   Returns true if authentication is successful, false otherwise.

   logout(): void

   Logs out the currently logged-in user.

   changePassword(username: string, oldPassword: string, newPassword: string): bool

   Allows the user to change their password.

   Returns true if password change is successful, false otherwise.

   viewMyAccount(username: string): UserProfile

   Returns the user profile associated with the given username.

2. **Admin Interface**

uploadCreditDataFile(file: File): void

Uploads a CSV file containing credit transaction data to the system.

viewUploadedFiles(): List<File>

Retrieves a list of all the credit data files uploaded to the system.

deleteUploadedFile(fileId: string): void

Deletes the file with the given fileId from the system.

analyseUploadedData(fileId: string): AnalysisReport

Analyzes the data from the uploaded file with the given fileId.

Returns an analysis report summarizing key insights and findings.

generateFilesReport(): List<AnalysisReport>

Generates reports based on the analysis of uploaded files.

Returns a list of analysis reports.

3. **Prediction Management Interface**

   makePredictionManually(formData: FormData): PredictionResult

   Allows users to input transaction data through a form to predict if it's fraudulent.

   Returns the prediction result.

   uploadSingleCSVRecord(file: File): PredictionResult

   Allows for uploading a single record in CSV format for fraud prediction.

   Returns the prediction result.

   uploadMultipleCSVRecords(files: List<File>): List<PredictionResult>

   Enables bulk upload of multiple records in CSV format for simultaneous fraud predictions.

   Returns a list of prediction results.

4. **Data And Storage Model Interface**

   getUserProfiles(): List<User>

   Retrieves all user profiles stored in the system.

   getTransactionData(): List<Transaction>

   Retrieves all transaction data stored in the system.

   getAnalysisReports(): List<Report>

   Retrieves all analysis reports stored in the system.

getPredictionResults(): Map<Transaction, Prediction>

Retrieves all prediction results stored in the system.

5. **System Interfaces Interface**

   getUserInterface(): UserInterface

   Retrieves the user interface.

   getAdminInterface(): AdminInterface

   Retrieves the admin interface.

   getDataProcessingInterface(): DataProcessingInterface

   Retrieves the data processing interface.

6. **Processes Interface**

   authenticateUser(username: string, password: string): bool

   Manages the login and logout functionalities, ensuring secure access to the system.

   Returns true if authentication is successful, false otherwise.

   handleData(file: File): void

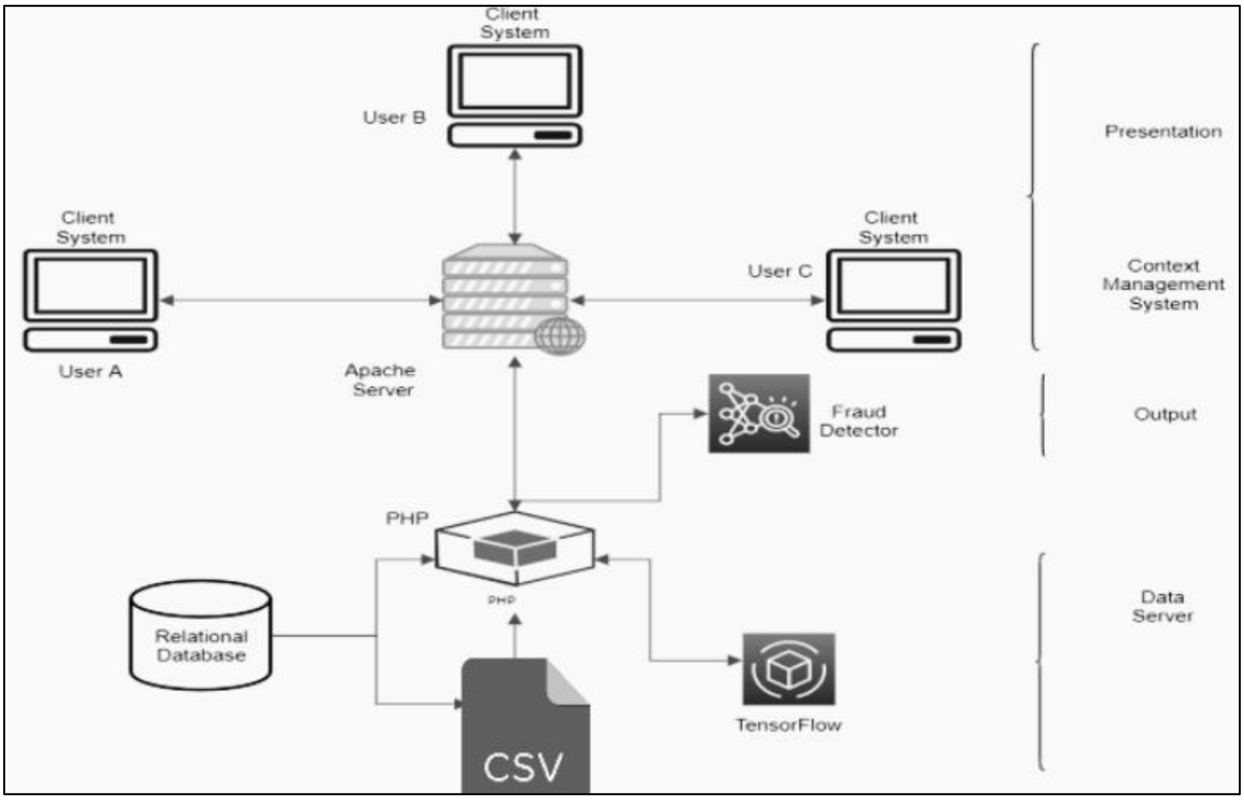
   Handles uploading, storing, deleting, and analyzing transaction data.

   makePrediction(data: Data): PredictionResult

   Employs machine learning models and rule-based systems to conduct fraud predictions based on uploaded or manually entered data.

   Returns the prediction result.

SYSTEM ARCHITECTURE



The web architecture is designed in such a way that we can use it as credit card fraud detection.

The web app was developed using python as server-side scripting and also the website will help the user to input the file with his details that will be read by the server and based on the details it will determine whether the transactions will have probability of fraud that has occurred.

We also made use of ML modules to perform the prediction and detecting the credit card frauds.

SYSTEM DESIGN

Frontend (User Interface)

Technologies:

- HTML
- CSS
- JavaScript
- Django

Components:

- User Login/Logout Pages
- Dashboard for transaction monitoring
- Forms for manual data entry and file uploads
- Admin panels for file and user management

Backend (Server-side Processing)

Technologies:

- Python

Components:

- Authentication Server: Handles login, logout, and security processes.
- Data Management Server: Manages file uploads, deletions, and data retrieval.
- Prediction Engine: Integrates machine learning models for fraud detection.
- Reporting and Analysis Module: Generates reports and performs data analytics.

Data Storage and Management

CSV File Storage:

- Store transaction data, user data, and results in separate CSV files.
- Organize files in a structured directory system for easy access and management.

File Management System:

- Develop a custom file management module to handle CRUD (Create, Read, Update, Delete) operations on CSV files.
- Use libraries such as Python's pandas for reading and writing CSV files efficiently.

Prediction Engine

Framework:

- Regression

Functionality:

- Train models on historical data to identify patterns.
- Predict fraud in real-time or batch processing modes.
- Continuously update models with new data for accuracy.

Component Interaction

Frontend to Backend:

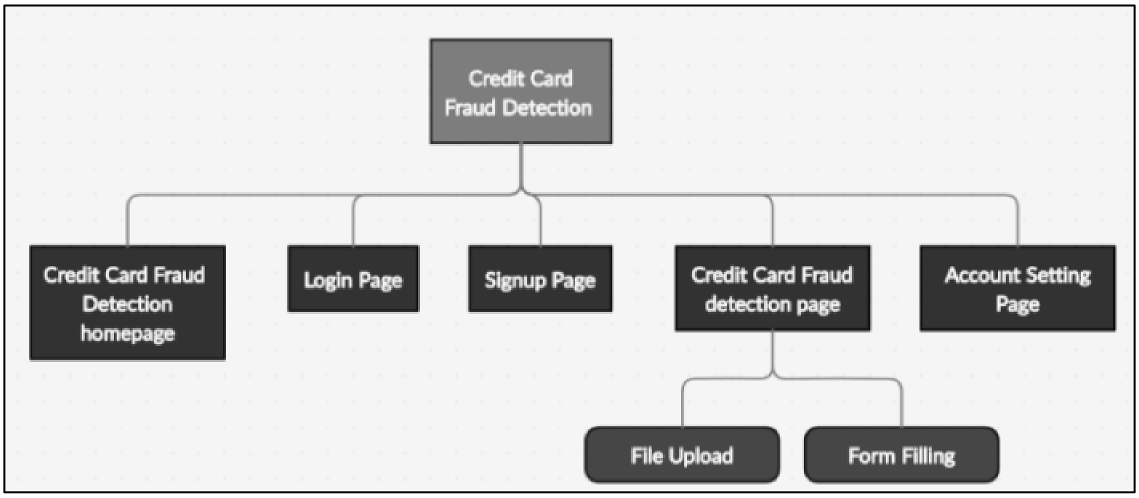
- Frontend sends requests to the backend for performing operations like user authentication, data upload, prediction requests, and report generation.

Backend to Machine Learning Models:

- Backend sends data to the machine learning service for fraud prediction and receives prediction results.

Backend to Database:

- Backend performs all data manipulation tasks including CRUD operations, and querying for analytics. Users to System: Users interact with the system through a web interface.


Credit Card Fraud Detection
Credit Card Fraud Detection homepage
Login Page
Signup Page
Credit Card Fraud detection page
Account Setting Page
File Upload
Form Filling

ALGORITHMS AND DATA STRUCTURES

1. Logistic Regression:

Algorithm: Logistic regression is a supervised learning algorithm used for binary classification tasks, like fraud detection.

Data Structure: Logistic regression doesn't require any specific data structure, but the data is typically organized in a tabular format where each row represents a transaction, and each column represents a feature (e.g., transaction amount, time, etc.).

2. Random Forest:

Algorithm: Random forest is an ensemble learning method that builds multiple decision trees and merges their predictions to improve accuracy and reduce overfitting.

Data Structure: Random forests work well with structured data represented in tabular format, similar to logistic regression.

3. Data Preprocessing:

Data preprocessing techniques like scaling, normalization, and handling missing values are important to prepare the data for both logistic regression and random forest.

4. Feature Engineering:

Creating new features from existing ones or transforming features can enhance the performance of both algorithms.

5. Model Evaluation:

Techniques like cross-validation, ROC curves, and precision-recall curves can be used to evaluate the performance of the models.

6. Handling Imbalanced Data:

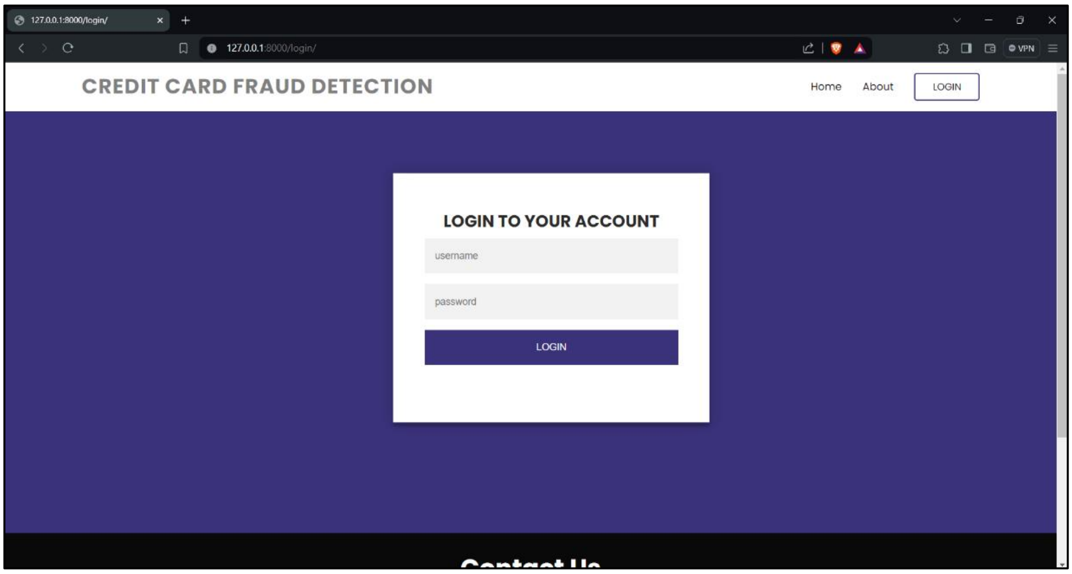
Credit card fraud datasets are usually highly imbalanced, with a small percentage of fraudulent transactions. Techniques like oversampling, undersampling, or using algorithms that handle class imbalance (e.g., weighted classes) are essential.

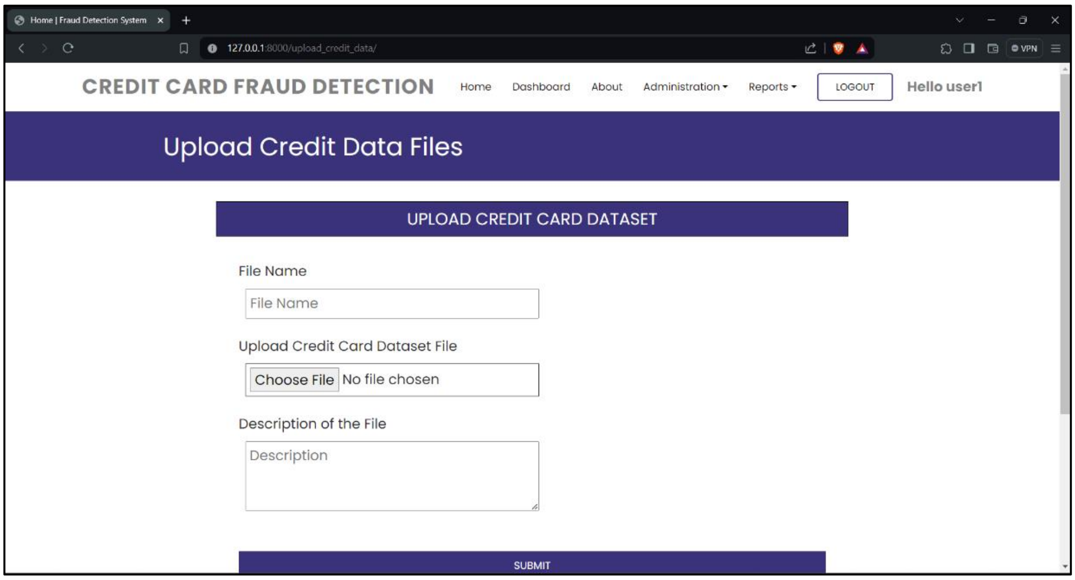
7. Model Deployment:

Once trained, the models need to be deployed into a production environment where they can be used to score incoming transactions in real-time or in batches.

By combining logistic regression and random forest within a comprehensive fraud detection system, you can leverage the strengths of both algorithms to improve detection accuracy and robustness.

USER INTERFACE AND IMPLEMENTATION




Home | Fraud Detection System
127.0.0.1:8000/upload_credit_data/
CREDIT CARD FRAUD DETECTION
Home
Dashboard
About
Administration
Reports
LOGOUT
Hello user1

Upload Credit Data Files

UPLOAD CREDIT CARD DATASET

File Name

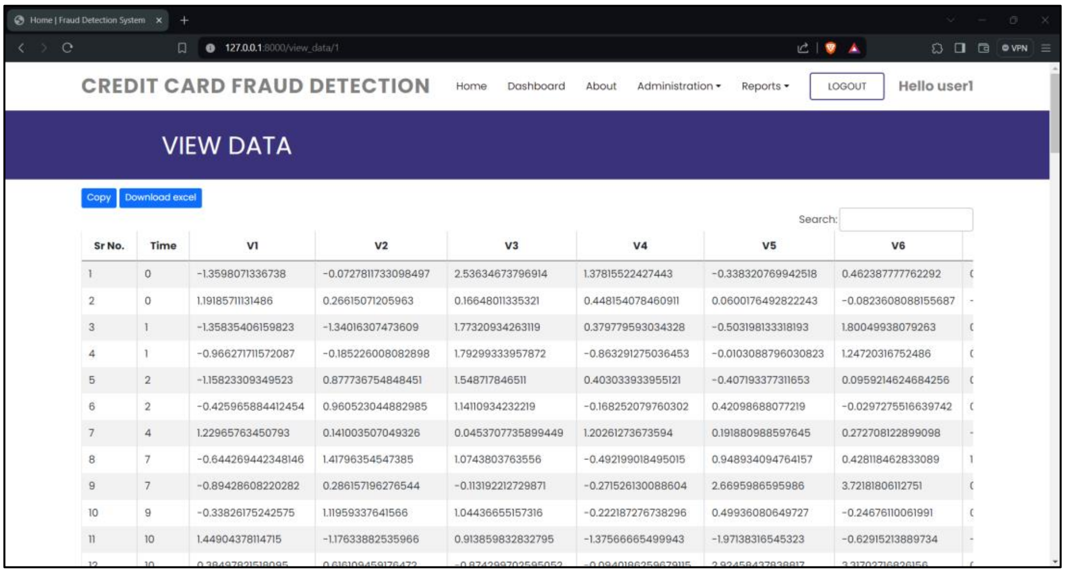
File Name

Upload Credit Card Dataset File

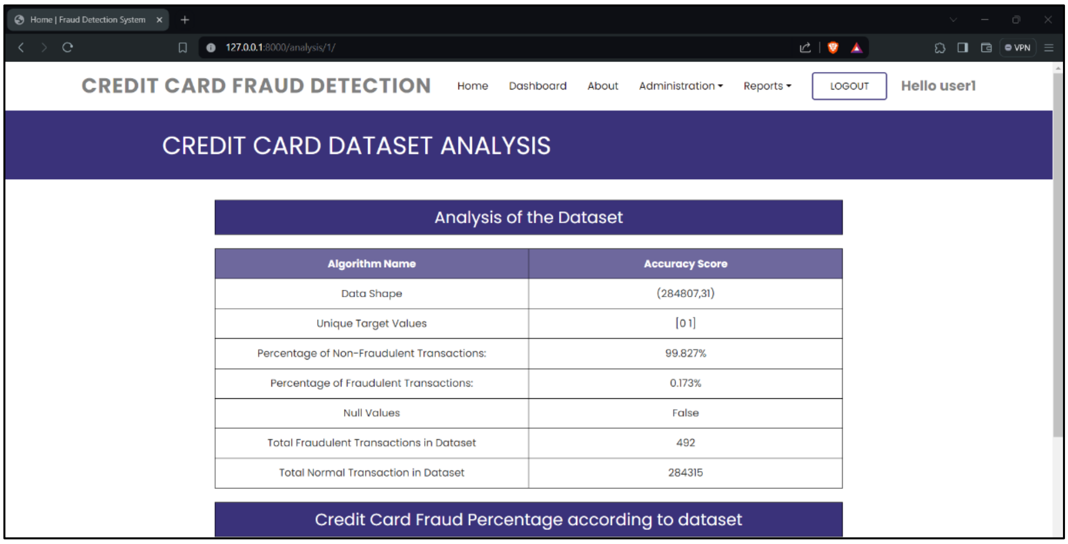
Choose File No file chosen

Description of the File

Description



Sr No.	Time	V1	V2	V3	V4	V5	V6
1	0	-1.3598071336738	-0.0727811733098497	2.53634673796914	1.37815522427443	-0.338320769942518	0.462387777762292
2	0	1.19185711131486	0.26615071205963	0.16648011335321	0.448154078460911	0.0600176492822243	-0.0823608088155687
3	1	-1.35835406159823	-1.34016307473609	1.77320934263119	0.379779593034328	-0.503198133318193	1.80049938079263
4	1	-0.966271711572087	-0.185226008082898	1.79299333957872	-0.863291275036453	-0.0103088796030823	1.24720316752486
5	2	-1.15823309349523	0.877736754848451	1.548717846511	0.403033933955121	-0.407193377311653	0.0959214624684256
6	2	-0.425965884412454	0.960523044882985	1.14110934232219	-0.168252079760302	0.42098688077219	-0.0297275516639742
7	4	1.22965763450793	0.141003507049326	0.0453707735899449	1.20261273673594	0.191880988597645	0.272708122899098
8	7	-0.644269442348146	1.41796354547385	1.0743803763556	-0.492199018495015	0.948934094764157	0.428118462833089
9	7	-0.89428608220282	0.286157196276544	-0.113192212729871	-0.271526130088604	2.6695986595986	3.72181806112751
10	9	-0.33826175242575	1.11959337641566	1.04436655157316	-0.222187276738296	0.49936080649727	-0.24676110061991
11	10	1.44904378114715	-1.17633882535966	0.913859832832795	-1.37566665499943	-1.97138316545323	-0.62915213889734
12	10	0.38497821518095	0.616109459176472	-0.874299702595052	-0.0940186259679115	2.92458437838817	3.31702716826156


Home | Fraud Detection System
127.0.0.1:8000/analysis/1/
CREDIT CARD FRAUD DETECTION
Home
Dashboard
About
Administration ▾
Reports ▾
LOGOUT
Hello user1
CREDIT CARD DATASET ANALYSIS

Analysis of the Dataset

Algorithm Name	Accuracy Score
Data Shape	(284807,31)
Unique Target Values	[0 1]
Percentage of Non-Fraudulent Transactions:	99.827%
Percentage of Fraudulent Transactions:	0.173%
Null Values	False
Total Fraudulent Transactions in Dataset	492
Total Normal Transaction in Dataset	284315

Credit Card Fraud Percentage according to dataset

DESIGN OF TESTS

Test Case Id	Test Objective	Precondition	Steps	Expected Result	Postcondition
TC_01	Data Collection	Admin uploads credit data	1. Login to the portal 2. Go to dashboard 3. Upload files	View uploaded data	Data stored in database
TC_02	Model Training	Existing training and test dataset	1. Open kernel 2. Load test and train dataset to kernel 3. Load model to kernel 4. Fit model and train	Accuracy prediction and classification result	Save model
TC_03	Model Testing	Existing trained model	1. Open kernel 2. Load trained model 3. Load test data 4. Review fraud prediction	Transaction fraud prediction	Chance of fraud detected
TC_04	Model Prediction	Existing trained model and existing UI for visualization	1. Open prediction UI (click on view analysis) 2. View prediction	Transaction fraud prediction and visualization	Chance of fraud detected

CONCLUSION

The proposed credit card fraud detection system is built upon a robust architecture that combines CSV files for storing and processing data with a comprehensive backend equipped with state-of-the-art machine learning models. This system is designed to effectively predict and detect fraudulent activities in credit card transactions.

The architecture of the system is meticulously crafted to ensure user-friendliness, security, and efficiency. It seamlessly integrates real-time and batch processing capabilities to meet diverse operational demands.

By leveraging the power of machine learning and efficient data handling, the system can analyze transaction data in real-time, enabling quick and accurate identification of potential fraudulent activities. Moreover, its user-friendly interface ensures that users can easily access and interact with the system, making it an invaluable tool for financial institutions and credit card companies in safeguarding against fraudulent transactions.