Credit Fraud Detection Using the Hidden Markov Model

Team Members:

Sergio Gabriel Jiawei Kun ; s_kun@mail.fhsu.edu

Lee Joseph Judkins ; ljjudkins@mail.fhsu.edu

Project Github: https://github.com/kunsergio117/CreditFraudDetectionHMM.git

Instructor: Prof. Hieu Vu

Revision History:

Part 1	07/10/2024

1. Analysis and Domain Modeling

1.1 Concept Definitions

The concept definitions for our **Credit Fraud Detection System** include the key objects and processes involved in identifying fraudulent transactions using machine learning algorithms. Below are the primary concepts:

- **User**: Represents individuals or entities using the system to submit transaction data, receive alerts, and generate fraud reports.
- **Transaction**: The financial data entered into the system, including amount, time, location, and type of transaction. It is the primary input for fraud detection.
- **Fraud Alert**: A notification generated by the system when suspicious or potentially fraudulent transactions are detected.
- **Report**: A document or summary of the results of fraud detection activities, which can be accessed and downloaded by users.
- **Fraud Detection Model**: The core model responsible for detecting fraudulent activity. In this system, it refers to the Hidden Markov Model (HMM) and Recurrent Neural Network (RNN) used for fraud analysis.
- **Admin**: Users with elevated privileges responsible for system maintenance and updates, including model retraining and user management.

1.2 Association Definitions

- **User - Transaction (submits)**: A user submits a series of transactions to the system for fraud analysis.
- **Transaction - Fraud Alert (triggers)**: When a transaction is flagged as suspicious, it triggers a fraud alert.
- **User - Fraud Alert (receives)**: Users are notified of any fraud alerts generated by their transactions.
- **Admin - Fraud Detection Model (manages)**: The admin manages and updates the fraud detection models used in the system.

1.3 Attribute Definitions

- **User Attributes**:
    - UserID: Unique identifier for each user.
    - UserName: Name of the user.
    - UserRole: Role of the user (e.g., customer, admin).
- **Transaction Attributes**:
    - TransactionID: Unique identifier for each transaction.
    - Amount: Monetary value of the transaction.
    - Timestamp: The time when the transaction occurred.
    - Location: Location of the transaction.
    - Status: Indicates whether the transaction is flagged as fraudulent or legitimate.

- **Fraud Alert Attributes**:
    - AlertID: Unique identifier for each alert.
    - AlertLevel: Severity of the fraud suspicion (low, medium, high).
    - TransactionID: Reference to the associated transaction that triggered the alert.
- **Report Attributes**:
    - ReportID: Unique identifier for the report.
    - CreationDate: The date the report was generated.
    - UserID: The user requesting the report.

2. Conceptual Model

The **conceptual model** represents the key entities and their associations in the system. Here is the domain model diagram:

[Insert Domain Model Diagram]
(The diagram would show entities like **User**, **Transaction**, **Fraud Alert**, **Fraud Detection Model**, and **Report**, along with their associations as defined above.)

3. Traceability Matrix

Below is the traceability matrix that maps the **use cases** to their corresponding **domain concepts**:

Use Case	User	Transaction	Fraud Alert	Fraud Detection Model	Report
Submit Transaction	X	X			
Receive Fraud Alert	X	X	X		
Generate Fraud Report	X				X
Manage Fraud Detection Model	X (Admin)			X	

4. System Operation Contracts

For the fully-dressed use cases identified in the earlier section, we define **system operation contracts** for the critical system operations:

- **SubmitTransaction()**:
  - **Precondition**: User is authenticated and logged in.
  - **Postcondition**: A new Transaction object is created and added to the transaction database. The Fraud Detection Model is invoked to analyze the transaction.
- **ReceiveFraudAlert()**:
  - **Precondition**: A Transaction has been processed.
  - **Postcondition**: If the transaction is flagged as suspicious, a new Fraud Alert is created and sent to the user.
- **GenerateFraudReport()**:
  - **Precondition**: User requests a report.
  - **Postcondition**: A Report object is generated and stored, containing the results of fraud detection for the specified transactions.
- **UpdateFraudModel()** (Admin Operation):
  - **Precondition**: Admin is authenticated and logged in.
  - **Postcondition**: The existing Fraud Detection Model is updated or retrained with new data.

5. Data Model and Persistent Data Storage

5.1 Persistent Data Storage

Yes, our system requires persistent data storage to maintain transaction records, fraud detection results, and user data between sessions.

5.2 Persistent Objects

The **persistent objects** in our system include:

- **User Data**: Information about the users of the system, such as login credentials and roles.
- **Transaction Records**: A history of all transactions submitted to the system, including metadata such as time and location.
- **Fraud Alerts**: A log of all alerts triggered by suspicious transactions.
- **Reports**: Fraud reports generated upon user request.

5.3 Storage Strategy

We use a **relational database** to store the persistent objects. The database schema is structured as follows:

- **Users Table**:

    - UserID (Primary Key)
    - UserName
    - UserRole
    - PasswordHash
- **Transactions Table**:
    - TransactionID (Primary Key)
    - UserID (Foreign Key)
    - Amount
    - Timestamp
    - Location
    - Status
- **FraudAlerts Table**:
    - AlertID (Primary Key)
    - TransactionID (Foreign Key)
    - AlertLevel
    - Timestamp
- **Reports Table**:
    - ReportID (Primary Key)
    - UserID (Foreign Key)
    - CreationDate

5.4 File Formats

Here's how the data for each object is stored in separate files:

1. **Users Data File (users.csv)**
    - **File Name**: users.csv
    - **Description**: Contains the data of all system users, including their roles and hashed passwords for authentication.

Format:
text
Copy code
UserID,UserName,UserRole,PasswordHash
101,john_doe,customer,sf93kfjs93f
102,jane_doe,admin,8sjdfkjs82
103,bob_smith,customer,a8sd7as6df

    -
    - **Attributes**:
        - UserID: Unique identifier for each user.
        - UserName: The username chosen by the user.
        - UserRole: The role of the user (e.g., customer, admin).
        - PasswordHash: Hashed version of the user's password for secure storage.

2. **Transactions Data File (transactions.csv)**
    - **File Name**: transactions.csv
    - **Description**: Logs all financial transactions submitted by users.

      Format:
      TransactionID,UserID,Amount,Timestamp,Location,Status
      201,101,150.50,2024-10-06 14:23:00,New York,legitimate
      202,102,500.00,2024-10-07 09:45:00,Los Angeles,suspicious
      203,101,20.00,2024-10-07 16:12:00,New York,legitimate

    -
    - **Attributes**:
        - TransactionID: Unique identifier for each transaction.
        - UserID: The ID of the user who made the transaction.
        - Amount: The monetary amount of the transaction.
        - Timestamp: Date and time when the transaction occurred.
        - Location: The geographic location where the transaction was made.
        - Status: Indicates whether the transaction is flagged as legitimate or suspicious.
3. **Fraud Alerts Data File (fraud_alerts.csv)**
    - **File Name**: fraud_alerts.csv
    - **Description**: Stores data about alerts generated when suspicious transactions are detected.

      Format:
      AlertID,TransactionID,AlertLevel,Timestamp
      301,202,high,2024-10-07 09:46:00
      302,203,low,2024-10-07 16:13:00

    -
    - **Attributes**:
        - AlertID: Unique identifier for each fraud alert.
        - TransactionID: The transaction associated with the fraud alert.
        - AlertLevel: The severity level of the alert (e.g., low, medium, high).
        - Timestamp: The date and time the alert was generated.
4. **Reports Data File (reports.csv)**
    - **File Name**: reports.csv
    - **Description**: Contains the fraud analysis reports generated for each user.

      Format:
      ReportID,UserID,CreationDate
      401,101,2024-10-07 17:00:00
      402,102,2024-10-07 18:15:00

        -

- **Attributes**:
  - ReportID: Unique identifier for each report.
  - UserID: The ID of the user who requested the report.
  - CreationDate: The date and time the report was generated.

6. Mathematical Model

Our **Credit Fraud Detection System** uses two main mathematical models:

6.1 Hidden Markov Model (HMM)

The HMM is used to model the sequential nature of user transactions. It works by analyzing the probability of a sequence of transactions to determine whether the current transaction deviates from typical behavior patterns, thus flagging potential fraud.

- **States**: Represent different transaction types or behaviors (e.g., normal, suspicious).
- **Transition Probabilities**: The probability of moving from one state (transaction type) to another.
- **Emission Probabilities**: The probability of observing certain features (transaction amount, location, etc.) given the current state.

6.2 Recurrent Neural Network (RNN)

The RNN analyzes sequences of user transactions by leveraging previous transactions to make predictions about whether a current transaction is fraudulent. The RNN is well-suited for temporal data due to its ability to maintain memory of prior events.

- **Input**: Transaction sequences.
- **Output**: Binary classification (fraudulent vs. legitimate).
- **Loss Function**: Cross-entropy loss is used to optimize the model's prediction accuracy.

Project Management

Both team members will be actively engaged in the development process, utilizing the GitHub repository to track progress and contributions clearly.

Responsibilities will be delegated based on each member's strengths and expertise, with a shared accountability structure ensuring that all aspects of the project are covered. This is because our group consists of only a pair and thus realistically we will both need input and validation from one another in all development areas.

Below is our projected milestones, with week 1 representing the week beginning in 23 Sept 2024.

Week	Task Description	Responsible Team Members
Week 1	Project kick-off meeting, define scope and objectives	Sergio Kun, Joseph Judkins
Week 2	Research HMM algorithms and relevant literature	Sergio Kun
	Familiarization with data sources and datasets	Joseph Judkins
Week 3	Develop initial design and architecture of the application	Sergio Kun, Joseph Judkins
Week 4	Implement CSV transaction data upload feature	Sergio Kun
	Initial development of the user authentication process	Joseph Judkins
Week 5	Develop the manual checking function for transaction validity	Sergio Kun
Week 6	Implement the alerting function for suspicious transactions	Joseph Judkins

Week	Task Description	Responsible Team Members
Week 7	Integrate the simulated fraudulent transaction feature	Sergio Kun
Week 8	Testing functionality and debugging	Sergio Kun, Joseph Judkins
Week 9	Develop and integrate reporting tools for transaction summaries	Joseph Judkins
	User interface refinement and user experience enhancements	Joseph Judkins
Week 10	Conduct user testing and gather feedback	Sergio Kun, Joseph Judkins
Week 11	Finalize features based on user feedback	Sergio Kun, Joseph Judkins
Week 12	Prepare project presentation and documentation	Sergio Kun, Joseph Judkins
Week 13	Project review and adjustments based on tutor feedback	Sergio Kun, Joseph Judkins
Week 14	Final project deployment and presentation to the class	Sergio Kun, Joseph Judkins

References