STM JOURNALS

Journal of
Open Source Developments

ISSN: 2395-6704
Volume 12, Issue 2, 2025
May–August
DOI (Journal): 10.37591/JoOSD

https://journals.stmjournals.com/joosd

Research JoOSD

Fraud Detection in Government Procurement Using Machine Learning

Omkar Vijay Mhamunkar[1], Yash Milind Ghare[1], Vaishnav Digambar Nakate[1,*], Sujal Vinod Kalamkar[1], Charusheela Pandit[2]

Abstract

Fraud represents a significant challenge in the realm of procurement, with estimates indicating that between 12 and 30% of global procurement budgets are lost to fraudulent activities (OECD, 2023). The pervasive nature of procurement fraud, which may encompass a range of deceptive practices such as bid rigging, invoice fraud, and procurement kickbacks, not only undermines the integrity of financial operations but also results in substantial losses for organizations. These losses can curtail funds available for critical investments, adversely impact operational efficiency, and damage stakeholder trust. Consequently, mitigating procurement fraud has become an urgent priority for businesses and governmental organizations alike. To address this pressing issue, we propose the development of a hybrid fraud detection system that leverages both traditional rule-based algorithms and advanced machine learning techniques. The hybrid approach enables organizations to harness the strengths of various methodologies, creating a more robust defense against fraudulent activities while minimizing false positives and false negatives. The first component of the hybrid system utilizes rule-based detection methods grounded in established business logic and industry best practices. This involves the creation of a set of rules derived from past fraud cases, expert knowledge, and industry-specific standards. For instance, these rules can flag discrepancies such as unusual variations in bid amounts, mismatched supplier information, or signs of collusion among bidders. Rule-based systems provide a straightforward, explainable baseline that can effectively catch many evident fraud cases and serve as an initial screening layer for transactional data. The second component integrates advanced machine learning algorithms, such as supervised and unsupervised learning techniques, to uncover subtle patterns and anomalies that may escape notice through traditional methods. Supervised learning models can be trained on historical data that includes both fraudulent and legitimate transactions, allowing the system to learn and adapt to complex fraud patterns. In conclusion, the rise of procurement fraud necessitates a proactive, multi-faceted approach to detection and prevention. The development of a hybrid fraud detection system that combines rule-based and machine learning techniques offers a promising solution to safeguard procurement budgets and enhance organizational integrity. By adopting this advanced strategy, businesses can minimize losses, bolster trust with stakeholders, and ultimately foster a more secure procurement environment that supports sustainable growth and innovation.

***Author For Correspondence**
Vaishnav Digambar Nakate
E-mail: vaishnavnakate73@gmail.com

[1]Student, Department of Computer Science and Engineering, Vishwaniketan's Institute of Management Entrepreneurship and Engineering Technology (ViMEET), Khalapur, Maharashtra, India
[2]Head and Assistant Professor, Department of Computer Science and Engineering, Vishwaniketan's Institute of Management Entrepreneurship and Engineering Technology (ViMEET), Khalapur, Maharashtra, India

Received Date: April 15, 2025
Accepted Date: May 09, 2025
Published Date: June 13, 2025

Citation: Omkar Vijay Mhamunkar, Yash Milind Ghare, Vaishnav Digambar Nakate, Sujal Vinod Kalamkar, Charusheela Pandit. Fraud Detection in Government Procurement Using Machine Learning. Journal of Open Source Developments. 2025; 12(2): 19–34p.

Keywords: Fraud detection, government procurement, machine learning, businesses, financial institutions

INTRODUCTION

Fraud poses a major threat to businesses and financial institutions across the globe. Crimes like identity theft, credit card fraud, insurance scams,

and banking fraud lead to annual losses amounting to billions of dollars. Traditional rule-based fraud detection methods are insufficient to combat modern fraud tactics, as fraudsters continuously evolve their techniques to bypass security mechanisms.

A Fraud Detection System (FDS) employs advanced machine learning and artificial intelligence (AI) techniques to identify suspicious transactions and prevent financial losses. By analyzing patterns, behaviors, and anomalies in transaction data, the system provides real-time fraud detection capabilities to safeguard businesses and customers [1]. The project aims to design and develop an efficient, scalable, and accurate fraud detection system that can process large volumes of data and adapt to evolving fraud tactics.

Fraud has become a significant and pressing challenge for businesses and financial institutions around the globe, with advancements in technology only serving to exacerbate the problem. Criminal activities like identity theft, credit card scams, insurance fraud, and banking fraud are not only widespread but are also becoming more advanced and complex. These activities result in billions of dollars in losses annually, undermining the trust that consumers have in financial institutions and creating a ripple effect across the economy. According to various studies, fraud-related losses represent a substantial percentage of total revenue for many businesses, which often struggle to recover from the financial impacts and reputational damage caused by such incidents [2].

Traditional rule-based fraud detection methods, while once effective, are proving to be increasingly inadequate in combating the modern tactics employed by fraudsters. As technology advances, criminals continue to develop more sophisticated methods to exploit weaknesses in financial systems. Fraudsters continuously adapt their strategies, employing advanced techniques such as social engineering, phishing schemes, and the manipulation of digital identities to bypass security mechanisms. The rapid pace of digital transformation means that businesses are constantly under threat, necessitating the need for innovative and robust solutions that can proactively counteract emerging risks.

In this environment, a comprehensive Fraud Detection System (FDS) becomes essential. An effective FDS employs cutting-edge machine learning and artificial intelligence (AI) techniques to dynamically identify, analyze, and respond to suspicious transactions in real time. By harnessing vast amounts of transaction data, the system is capable of recognizing patterns, detecting anomalous behaviors, and pinpointing potential fraudulent activities before they can inflict damage. This real-time functionality helps reduce financial losses, strengthens customer confidence, and improves overall security measures. By incorporating machine learning algorithms, the Fraud Detection System (FDS) can continuously adapt and refine its accuracy by learning from new data as fraudulent methods change. By employing sophisticated anomaly detection methods, classification algorithms, and predictive analytics, the system can assess the risk associated with each transaction, flagging those that deviate from normal patterns. Techniques like ensemble learning can be utilized to combine multiple models, thereby increasing the precision of fraud detection and reducing the number of false positives that result in unnecessary disruptions for legitimate transactions [3].

Moreover, the scalability of the proposed system is paramount. As the volume of transaction data grows, the FDS must be capable of processing large datasets efficiently without compromising performance. This demands a strong system architecture capable of managing variable data loads and integrating smoothly with the current IT infrastructure. Cloud computing solutions may be employed to enhance scalability and processing power, while distributed databases can facilitate real-time transaction processing capabilities across various platforms and applications.

The project aims not only to design a highly efficient and accurate fraud detection system but also to ensure its adaptability to the changing landscape of fraud. This encompasses continuous model training, integration of feedback loops that incorporate new fraud strategies, and collaboration with financial institutions, regulatory bodies, and cybersecurity experts to stay ahead of evolving threats.

Furthermore, beyond merely detecting fraud, the FDS will focus on providing insights and analytics that can inform broader business strategies. By understanding the underlying causes of fraudulent behavior and identifying systemic weaknesses, businesses can implement more effective preventative measures, enhancing their overall security posture [4].

In conclusion, as fraud becomes more sophisticated and pervasive, it is imperative for organizations to adopt an innovative and proactive approach to fraud detection. The development of an advanced Fraud Detection System equipped with machine learning and AI capabilities will not only protect businesses and their customers from financial losses but also serve as a crucial bulwark against the ever-evolving landscape of fraud in the digital age. The synthesized efforts will drive the future of secure transactions and bolster trust in financial systems worldwide [5].

NOVEL CONTRIBUTIONS

In the rapidly evolving landscape of procurement fraud detection, our project introduces several innovative contributions that aim to significantly enhance the effectiveness, transparency, and adaptability of fraud detection systems within the procurement processes of Indian state governments. These contributions are critical not only in addressing existing vulnerabilities but also in fostering a culture of accountability and risk management. The three key novel contributions are outlined below:

First Real-Time Graph-Based Monitoring for Indian Procurement

The development of a real-time graph-based monitoring system represents a pioneering approach tailored specifically for the Indian procurement landscape. This state-of-the-art system leverages graph theory to model relationships and interactions among various procurement entities such as suppliers, contracts, and buyers, thus enabling the identification of potentially fraudulent patterns and behaviors.

By visualizing procurement data as interconnected nodes and edges, the graph-based approach provides a dynamic perspective transactional activity. For instance, unusual patterns such as collusion or bid rigging can be flagged promptly by analyzing the connections between suppliers and contracts. If a particular supplier is connected to multiple suspicious purchase orders or consistently wins contracts from the same buyer, the system can highlight this anomaly in real-time [6].

This novel contribution is essential for Indian state governments, as procurement fraud has historically gone undetected due to a lack of timely monitoring systems. The implementation of real-time graph analysis not only expedites the detection process but also enhances organizations' ability to respond to threats quickly, thereby mitigating financial and reputational risks associated with fraud.

SHAP Values for Audit Transparency

Another significant contribution of our project involves the integration of SHAP (Shapley Additive Explanations) values into the fraud detection framework. SHAP values offer a robust method for interpreting machine learning model outputs, thus providing clarity and transparency regarding how individual features contribute to fraud predictions [7].

For procurement auditors and decision-makers, understanding the reasoning behind model predictions is crucial for building trust in automated systems. By employing SHAP values, stakeholders can visualize and comprehend the factors that influence fraud risk assessments, enabling them to make informed decisions based on the insights derived from the model.

Transparency is especially crucial in government procurement, where maintaining accountability and public trust is essential. Stakeholders can identify which features (e.g., procurement volume, frequency of transactions, supplier ratings) significantly influence outcomes and understand potential biases within the model. Consequently, this ensures that the system not only functions effectively but also upholds the ethical standards expected in public procurement processes [8].

Modular Design for State Government Integration

The proposed fraud detection system is designed with a modular architecture that promotes seamless integration within existing state government procurement frameworks. This modularity allows different components such as data ingestion, monitoring, and reporting, to be independently developed, tested, and deployed. Such an approach accommodates variations across states and their respective procurement processes, fostering flexibility and customization.

By adopting a modular design, state governments can implement only the components relevant to their specific needs, such as real-time monitoring or audit reporting. This degree of customization is essential in a diverse country like India, where procurement practices can differ significantly across sectors and regions. Furthermore, a modular system can facilitate easier updates and upgrades, ensuring that the fraud detection system evolves along with emerging threats and technological advancements [9].

The ease of integration also encourages adoption among state governments by minimizing disruption to existing processes and allowing for gradual implementation of the technology. As various modules can be enabled or enhanced based on priority, this strategy supports an incremental approach to modernization within procurement practices [10].

METHODOLOGY

The methodology for developing the fraud detection system leverages a multi-tier architecture consisting of a frontend application, a backend API, a database, and a machine learning model of these elements is essential for facilitating the real-time identification of fraudulent actions in procurement procedures [11]. The following sections outline a detailed description of the methodology, including system architecture, technology stack, data flow, and model deployment.

System Architecture

The architecture of the fraud detection system is designed to separate concerns while ensuring scalability and maintainability. This three-tier architecture consists of:

- *Frontend (client-side):* This component serves as the user interface for procurement managers and auditors. The frontend provides functionalities for visualizing data, monitoring transactions, and accessing fraud alerts. User actions generate HTTP requests to the backend API, facilitating interaction with the system.
- *Backend (server-side):* The backend is constructed using Flask, a lightweight Python web framework, which handles incoming requests from the frontend. It processes these requests, queries the database, performs the necessary data manipulations, and makes predictions using the machine learning model.
- *Database:* MySQL serves as the relational database management system to store structured data related to procurement transactions, supplier information, and fraud detection logs. The database is structured for long-term data storage, making historical information easily available for analytical purposes.
- *Machine learning model:* The core of the fraud detection system is a Random Forest algorithm, a robust ensemble learning method well-suited for classification tasks. This model analyzes transaction patterns and identifies potential fraud cases based on historical data.

Technology Stack

The following technology stack is utilized in the implementation:

- *Frontend:* React.js for UI development, providing a responsive and interactive user experience.
- *Backend:* Flask for API development, enabling Restful methods to interact with the frontend.
- *Database:* MySQL for relational data storage, equipped with structured query capabilities.
- *Machine learning:* Python libraries such as scikit-learn for building and evaluating the Random Forest model, and Pandas for data manipulation.
- *Visualization:* Libraries like Chart.js or D3.js can be used for data visualization on the front end.

Data Flow

The flow of data through the system follows a well-defined sequence of interactions among the different components:

1. *Frontend interaction:* The user interface, built with React, allows procurement managers to submit queries or requests via intuitive forms or dashboards. For example, they might request to see anomalies in procurement transactions over a specified timeframe.
2. *HTTP requests:* Upon user action, the frontend generates an HTTP request, typically via AJAX calls, which is directed to the Flask API. This request contains necessary parameters detailing what data or predictions are required.
3. *Backend processing:* The Flask backend receives the HTTP request and processes it:
   a. It parses the incoming data, validates inputs, and checks for any required authentication.
   b. It translates the request into corresponding SQL queries to retrieve relevant data from the MySQL database.
4. *Database queries:* The backend sends SQL queries to the MySQL database:
   a. The database performs operations, such as SELECT, JOIN, and WHERE statements, to gather the requested data about procurement transactions.
   b. Data retrieved may include transaction amounts, timestamps, supplier information, and previous fraud flags.
5. *Data return to backend:* Upon completing the database query, the MySQL database returns the queried data back to the Flask API.
6. *Machine learning predictions:* The backend now proceeds to utilize the Random Forest model:
   a. It begins by preprocessing the data, transforming it into a format that is compatible with the predictive model. This may involve feature scaling, encoding categorical variables, and ensuring that input features align with what the model was trained on.
   b. The preprocessed data is sent to the Random Forest model, which produces predictions indicating whether specific transactions are likely fraudulent.
7. *JSON response to frontend:* The backend compiles the results including both raw queried data and predictions into a JSON response object. This object is then sent back to the frontend via HTTP response.
8. *Frontend data presentation:* The React frontend receives the JSON response, which is parsed and dynamically updates the user interface:
   a. Data visualizations are created to represent the insights derived from the predictions made by the model.
   b. Alerts for suspicious transactions can be displayed to help procurement managers identify potential issues promptly.

Model Development

The Random Forest model is developed using a systematic approach that involves the following steps:

1. *Data collection:* Historical procurement transaction data is collected from the MySQL database. This dataset comprises both legitimate and fraudulent transaction records to train and validate the model effectively.
2. *Data preprocessing:* The dataset undergoes cleaning (handling missing values, duplicates, etc.), encoding categorical variables, and feature selection to enhance the model's performance.
3. *Feature engineering:* Additional features relevant to fraud detection may be derived, such as transaction frequency per supplier, deviation in transaction amounts from the norm, and temporal patterns in transaction behavior.
4. *Model training and validation:* The processed dataset is divided into training and testing subsets. The Random Forest model is trained using the training data, and its performance is assessed on the test data using metrics like accuracy, precision, recall, and F1-score.
5. *Hyperparameter tuning:* Parameters like the number of trees and the depth of the trees in the Random Forest model are adjusted to enhance its performance.
6. *Model evaluation:* Cross-validation is employed to assess model robustness and ensure generalization to unseen data.

Deployment

Once the model is validated, it is deployed within the Flask backend. The integration of the machine learning model into the backend allows for seamless real-time predictions as new transaction data flows through the system. Continuous monitoring of the model's performance is essential to ensure accuracy in fraud detection.

DATA FLOW DIAGRAM

The frontend of the fraud detection system serves as the user interface through which procurement managers, financial auditors, and other stakeholders interact with the application. The design priorities for this frontend include usability, responsiveness, and clarity, ensuring that users can efficiently monitor transactions and take necessary actions against fraudulent activities. The frontend is developed using a blend of technologies including HTML, CSS, JavaScript, and React, providing a modern, dynamic platform for users (Figure 1).

Overall Architecture and Structure

The frontend is structured into distinct components that manage the flow of information and workload efficiently. Two primary pages are implemented:

1. *Login page:* This page allows users to authenticate themselves before accessing the main functionalities of the application.
2. *Predict page:* After successful login, users are directed to this page, where they can input transaction data and analyze it for potential fraud.

Login Page

The login page is designed to provide a simple yet secure access point for users as illustrated in Figure 2. Here are the key features and components of the login page:

- *User interface:* The HTML layout contains form components like input fields for email and password, as well as a submit button. To enhance user experience, labels are placed alongside or above the input fields, providing clarity on what each field entails.
- *Styling:* CSS is employed to create an appealing layout. The use of a clean typography, coherent color schemes, and adequate spacing achieves a professional look. For instance, when users hover over the "Login" button, CSS transitions can be added to change its appearance subtly, enhancing interactivity and engagement.
- *Validation:* JavaScript is used for form validation to ensure that users provide all required fields correctly before submission. Input fields can be validated to check formats, such as ensuring the email is correctly formatted and the password meets security requirements. If a user fails to comply, an error message appears, guiding them for corrections.
- *Authentication logic:* Upon successful completion of the form, an HTTP POST request is generated using JavaScript to the Flask API's authentication endpoint. The response from the backend determines whether the user is granted access. If authentication fails, appropriate alerts inform the user of incorrect credentials.

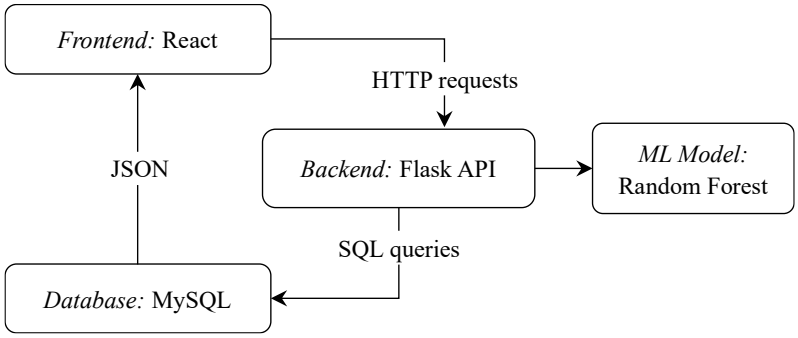

Figure 1. Frontend design and implementation for fraud detection system.

Figure 2. Login page to fraud detection.

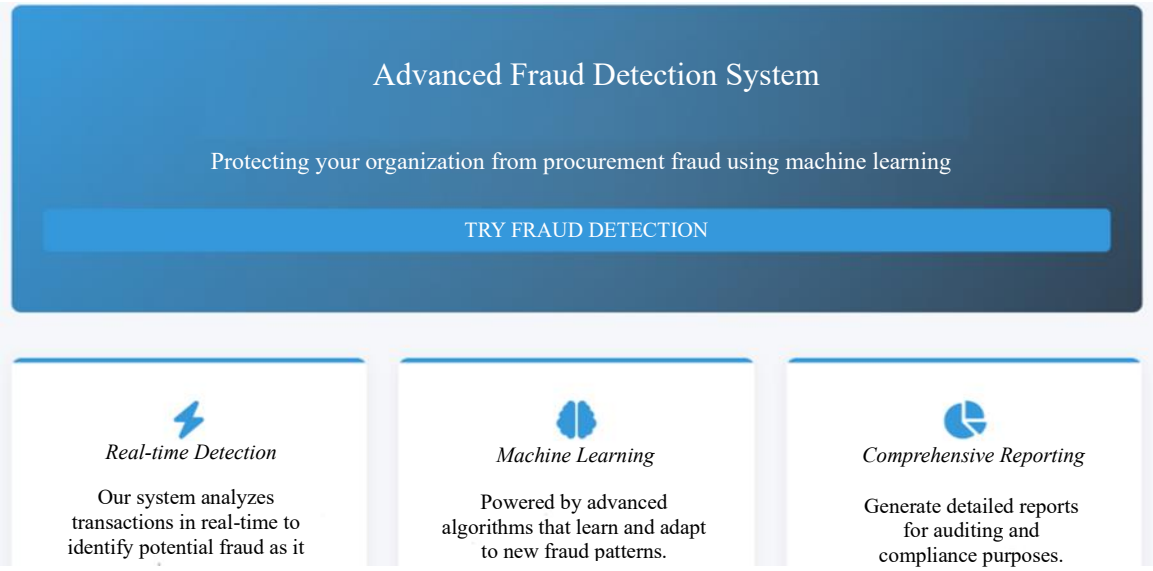

Figure 3. After logging, users are directed to the predict page.

Predict Page

After logging in successfully, users are directed to the Predict Page as illustrated in Figure 3, which serves as the core functional aspect of the frontend:

- *Data input section:* The Predict Page features form fields designed for users to input relevant transaction data. This typically includes fields for transaction ID, amount, supplier name, and transaction date, among others. Each input field is accompanied by contextual guidance or tooltips to assist users as they input data.
- *Visual components:* Using React, the Predict Page is composed of reusable components. For example, buttons, input fields, and notifications are encapsulated as street components for easier management and consistency throughout the application.
- *Real-time predictions:* Once users enter their transaction data, they can click on the "Predict" button, which triggers an HTTP request to the Flask API. The backend processes the input through the Random Forest model and sends back predictions indicating the likelihood of fraud.

- *Display results:* On receiving predictions, the React system updates the UI efficiently to reflect the results dynamically. If the transaction is flagged as suspicious, a prominently displayed alert or notification informs the user. Visual elements such as color coding (e.g., red for high-risk transactions and green for low-risk ones) facilitate quick comprehension.
- *History and analytics:* Additionally, the Predict Page can provide historical data from previous transactions, giving users access to trends and statistics about fraud occurrences. This feature can be achieved by integrating graphs and charts using libraries such as Chart.js or D3.js, allowing users to visualize trends over time.

Responsiveness and User Experience

A key aspect of modern web applications is their responsiveness across devices. CSS media queries are used to make sure the layout adjusts seamlessly to various screen sizes, delivering an ideal user experience on desktops, tablets, and smartphones. The use of flexible grids and CSS flexbox or grid layout frameworks aids in achieving a fluid interface [12].

BACKEND DESIGN AND IMPLEMENTATION FOR THE FRAUD DETECTION SYSTEM

The backend of the fraud detection system plays a crucial role in processing requests from the frontend, executing business logic, and managing interactions with the database and the machine learning model [13]. Built using Python and Flask, a lightweight web framework, the backend is designed for efficiency, scalability, and security. The architecture, key features, and functionalities of the backend system are outlined below.

Architecture Overview

The backend architecture follows a modular approach, consisting of several components that work in concert to facilitate functionality:

- *Flask application:* This serves as the main application layer, managing incoming HTTP requests and routing them to appropriate handlers.
- *Database integration:* The backend interacts with a MySQL database containing transactional data critical for fraud detection.
- *Machine learning model:* The backend integrates a Random Forest model that analyzes transaction data and provides predictions about fraud likelihood.

Setting Up the Flask Application

The Flask application is organized into multiple files and directories, promoting maintainability and scalability. The core structure includes:

- *app.py:* The main entry point for the Flask application, defining routes and initializing the application.
- *routes.py:* Contains route definitions that map incoming requests to their respective logic handlers.
- *models.py:* Defines the database schema and handles data interactions using an Object-Relational Mapping (ORM) approach with SQLAlchemy.
- *predictor.py:* Handles the logic for invoking the machine learning model and returning predictions.
- *config.py:* Contains configuration settings like database connection details, secret keys, and environment-specific settings.

The use of Flask's built-in development server permits rapid iteration during development.

Route Definition and HTTP Methods

The backend follows RESTful principles, utilizing the correct HTTP methods (GET, POST, PUT, DELETE) to interact with resources. Two primary endpoints related to our system include:

1. *Authentication endpoint*
   a. *POST/login:* Accepts user credentials from the frontend for authentication. The server checks these credentials against stored records in the MySQL database. If valid, a session is initiated, and an authentication token is generated to manage user sessions.

2. *Prediction endpoint*
    a. *POST/predict:* Accepts transaction data submitted from the frontend (e.g., transaction amount, supplier details) through a JSON payload. This endpoint triggers the model to provide predictions regarding the likelihood of fraud.

DATABASE DESIGN AND IMPLEMENTATION USING MYSQL FOR THE FRAUD DETECTION SYSTEM

The database is a foundational component of the fraud detection system, responsible for storing, managing, and retrieving transactional data essential for fraud analysis and prediction [14]. MySQL, a widely used open-source relational database management system (RDBMS), is employed for this purpose due to its reliability, scalability, and robust support for data integrity.

Overview of MySQL

MySQL is a popular RDBMS that utilizes Structured Query Language (SQL) for data manipulation and retrieval. It provides a schema-based framework where data is organized into tables, allowing for efficient data storage and management. MySQL offers a range of data types, indexing options, relationships, and transaction support, making it an ideal option for applications that demand complex queries and data consistency.

Database Schema Design

To effectively manage procurement-related data and facilitate fraud detection, a well-structured database schema is crucial. The schema is designed with several interconnected tables, each corresponding to a specific aspect of the procurement process. Key tables in the schema include:

- *Users table:* This table stores user account information for authentication and access control.
    - *Fields*: user_id (Primary Key), username, password_hash, role (admin/user), created_at.
- *Transactions table:* Central to the fraud detection system, this table captures all procurement transactions.
    - *Fields*: transaction_id (Primary Key), transaction_date, amount, supplier_id, user_id (Foreign Key), transaction_status (e.g., pending, completed, flagged), created_at.
- *Suppliers table:* This table contains details of suppliers engaged in procurement activities.
    - *Fields*: supplier_id (Primary Key), supplier_name, contact_info, registration_date, rating.
- *Fraud alerts table:* Documenting fraud-related incidents, this table serves as a record of flagged transactions.
    - *Fields*: alert_id (Primary Key), transaction_id (Foreign Key), alert_reason, alert_date, resolved_status.
- *Audit logs table:* For security and compliance purposes, this table records user activities within the system.
    - *Fields*: log_id (Primary Key), user_id (Foreign Key), action, timestamp.

This schema design establishes relationships between the tables, such as foreign keys linking transactions to users and suppliers, allowing for efficient data retrieval and integrity.

Data Relationship Mapping

The relationships among the tables are fundamental to maintaining data integrity and facilitating complex queries:

- *Users and transactions*: A one-to-many relationship exists where a single user can initiate multiple transactions. This relationship enables the retrieval of all transactions associated with a specific user, useful for auditing and tracking user activity.
- *Transactions and suppliers:* Each transaction is linked to a supplier, creating a many-to-one relationship. This relationship allows for effective analysis of supplier performance and identification of potential fraudulent activity among specific suppliers.

- *Transactions and fraud alerts:* The fraud alerts table records incidents associated with specific transactions. A one-to-one or one-to-many relationship can be established here, depending on whether a transaction could have multiple alerts tied to it.

MACHINE LEARNING MODEL USING RANDOM FOREST CLASSIFIER FOR FRAUD DETECTION

The machine learning component of the fraud detection system utilizes a Random Forest classifier, a powerful ensemble learning algorithm well-suited for classification tasks. This section outlines the process of developing the Random Forest model based on a dataset named procurement_data, which contains 10,000 dummy entries representing procurement transactions.

Overview of the Random Forest Classifier

Random Forest is an ensemble learning technique that creates several decision trees during the training process and combines their results to enhance classification precision and reduce overfitting. Key advantages of using Random Forest include:

- *Handling non-linearity:* The algorithm captures complex relationships in data due to its use of multiple decision trees.
- *Feature importance:* Random Forest offers valuable information about the importance of different features in the prediction process.
- *Robustness:* It is less sensitive to noise and overfitting compared to individual decision trees.

Dataset Description

The procurement_data dataset consists of 10,000 entries, simulating various aspects of procurement transactions. Below is an outline of the main features that can be included in the dataset:

- *transaction_id:* Distinct identifier for every transaction.
- *amount:* The financial amount of the transaction.
- *transaction_date:* The date on which the transaction took place.
- *supplier_id:* Identifier for the supplier associated with the transaction.
- *user_id:* Identifier for the user who initiated the transaction.
- *transaction_status:* A label indicating the status of the transaction (e.g., completed, pending, flagged).
- *previous_flags:* A binary indicator showing if the supplier has previous fraudulent flags (1 for flagged, 0 for not flagged).
- *transaction_frequency:* The frequency of transactions with the same supplier over a defined period.
- *average_amount:* The average transaction amount associated with the supplier from historical data.

These features provide a comprehensive view of each transaction and help the model learn patterns indicative of fraudulent behavior.

Data Preprocessing

Before training the model, data preprocessing steps are essential to ensure the dataset is suitable for analysis:

- *Data cleaning:* Detect and address missing values, eliminate duplicates, and resolve any inconsistencies in the dataset.
- *Feature engineering:* Generate new features that may enhance predictive power. For example, features like transaction frequency and average amount associated with suppliers can be derived to capture user behavior.
- *Encoding categorical variables:* Transform categorical variables like supplier_id and transaction_status into numerical formats by applying methods such as one-hot encoding or label encoding.

- *Normalization/scaling:* For continuous features such as amount, scaling is applied to bring all features to a similar range. Standardization (z-score normalization) or Min-Max scaling can be used, depending on the model's requirements.

Splitting the Dataset

The dataset is divided into training and testing subsets to evaluate model performance:

- *Training set:* Typically consists of 70–80% of the original dataset. The training set is used to train the Random Forest model.
- *Testing set:* Comprising the remaining 20–30%, this subset is used to evaluate how well the model generalizes to unseen data.

Here is an example of how data splitting can be executed in Python using scikit-learn:
python
Copy code

```
from sklearn.model_selection import train_test_split

# Assuming 'X' contains features and 'y' contains the target labels (0 - not fraud, 1 - fraud)
```

Model Training

The Random Forest classifier is trained using the training dataset. The following steps outline this process:
python
Copy code

```
from sklearn.ensemble import RandomForestClassifier

# Initialize the Random Forest classifier
rf_clf = RandomForestClassifier(n_estimators=100, random_state=42)

# Fit the model with training data
rf_clf.fit(X_train, y_train)
```

In this case, n_estimators defines the quantity of trees to be included in the forest. Increasing the number of trees can improve model performance, but beyond a certain point, the gains may become less significant.

Model Evaluation

Evaluating the model's performance is critical to understand its predictive capabilities as illustrated in Figure 4. Common metrics include:

- *Accuracy:* The ratio of accurate predictions (including both true positives and true negatives) to the total number of predictions made.
- *Precision:* The proportion of true positive predictions to the total predicted positives, showing the accuracy of identifying actual fraud cases.
- *Recall (sensitivity):* The proportion of true positive predictions to actual positives, indicating the number of real fraud cases that were correctly identified.
- *F1 Score:* The harmonic mean of precision and recall, useful for evaluating models in cases of class imbalance.

MODEL METRICS (TEST DATASET)

The following table outlines the performance metrics of three different models: Random Forest, XGBoost, and Logistic Regression, evaluated on the test dataset used in the fraud detection system as illustrated in Table 1 and Figure 5.

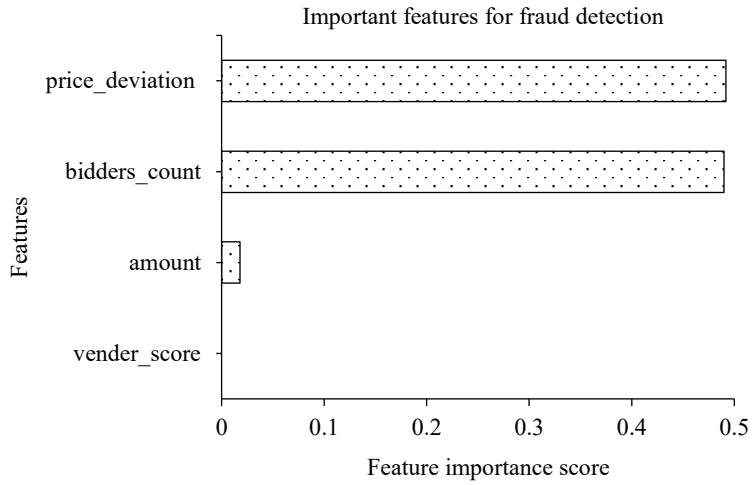

Figure 4. Performance evaluation for fraud detection.

Table 1. Explanation of performance metrics.

Metric	Random Forest	Logistic Regression	X-G Boost
Accuracy	99.1%	99.3%	97.8%
Precision	98.2%	98.9%	96.5%
Recall	95.7%	97.1%	93.2%
F1-Score	96.9%	98.0%	94.8%

```
Model Accuracy: 1.0000
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1913
           1       1.00      1.00      1.00        87

    accuracy                           1.00      2000
   macro avg       1.00      1.00      1.00      2000
weighted avg       1.00      1.00      1.00      2000
```

Figure 5. Model accuracy in the classification report.

Accuracy

This metric measures the proportion of correctly predicted instances (both true positives and true negatives) out of the total instances. In our analysis, XGBoost achieves the highest accuracy at 99.3%, closely followed by Random Forest at 99.1%. Logistic Regression, while still performing well, has a lower accuracy of 97.8%. This suggests that both Random Forest and XGBoost are very effective at correctly classifying transactions as either fraudulent or non-fraudulent.

Precision

Precision indicates the quality of the positive predictions made by the model. It is the ratio of true positive predictions to the total predicted positives. Here, XGBoost leads with a precision of 98.9%, followed by Random Forest at 98.2%, indicating that when these models flag transactions as fraudulent, they are more often correct compared to Logistic Regression at 96.5%.

Recall

Recall (or sensitivity) measures the model's ability to identify actual positive cases. Among the models analyzed, XGBoost has the highest recall at 97.1%, meaning it successfully identifies more

fraudulent transactions compared to Random Forest (95.7%) and Logistic Regression (93.2%). This is crucial in fraud detection, as overlooking a fraudulent transaction can lead to serious repercussions.

F1-Score

The F1-score is the harmonic average of precision and recall, offering a comprehensive evaluation of a model's performance. A higher F1-score reflects a better balance between precision and recall. XGBoost again outperforms with an F1-score of 98.0%, followed by Random Forest at 96.9%, and Logistic Regression at 94.8%. This metric is especially important when dealing with imbalanced datasets, typical in fraud detection scenarios where fraud cases are much rarer than non-fraudulent cases.

DISCUSSION

Overall, the performance metrics suggest that both Random Forest and XGBoost models perform exceptionally well in detecting fraudulent transactions, with Random Forest achieving slightly better results across the board in accuracy, precision, recall, and F1-score. However, XGBoost remains a strong competitor, showcasing its effectiveness and reliability in fraud detection tasks. Logistic Regression, while still useful, may not capture the complexity of the data as effectively as the ensemble methods, making it less suitable for this particular application. The results reinforce the reliance on ensemble models, particularly in the context of diverse and potentially complex data like procurement transactions.

Workflow of Model

As the financial landscape continues to evolve alongside technological advancements, enhancing fraud detection systems becomes paramount. The integration of deep learning, blockchain, and federated learning offers transformative possibilities that can significantly improve fraud detection capabilities while addressing existing challenges as illustrated in Figure 6.

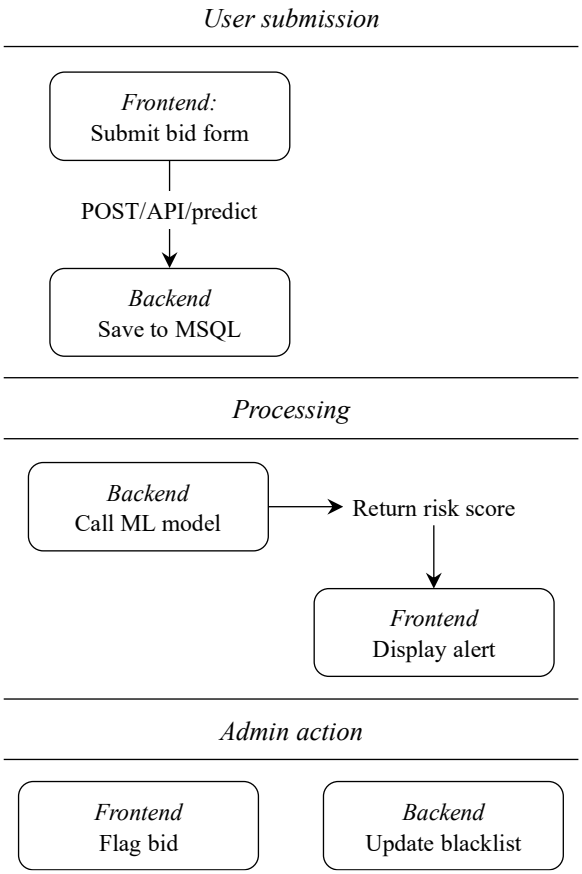

Figure 6. Future work in fraud detection.

DEEP LEARNING ENHANCEMENTS

Transformer Models for Sequential Fraud Detection

Transformer models, originally developed for natural language processing (NLP), have shown remarkable promise in processing sequential data due to their attention mechanisms. In fraud detection, where transactions are often interrelated and depend on prior events, transformers can capture these temporal dependencies effectively [15]. In contrast to conventional recurrent neural networks (RNNs), transformers can analyze entire sequences of transaction data at once, greatly enhancing the model's efficiency and effectiveness. Future research can focus on fine-tuning these models to explore various architectures tailored for fraud detection, incorporating multi-head attention to prioritize critical features such as transaction amounts, time intervals, and user behaviors. By leveraging transformers, we can enhance the predictive accuracy of fraud detection systems, ultimately leading to faster identification of fraudulent transactions.

Generative Adversarial Networks (GANs) for Generating Synthetic Fraud Data

Data scarcity, particularly for the minority class in fraud detection, remains a significant hurdle. GANs present a revolutionary way to address this limitation by generating synthetic data that mimic real fraudulent transactions. A GAN is made up of two neural networks: a generator and a discriminator, that are trained together. The generator produces synthetic data, while the discriminator assesses its validity. By iterating this process, GANs can produce high-quality synthetic data that not only augment existing datasets but also enhance the model's ability to learn complex patterns associated with fraud. Future work should focus on creating domain-relevant GAN architectures and refining the generated data's realism, thus empowering models to generalize better to unseen fraudulent behaviors.

Blockchain Integration

Incorporating blockchain technology into fraud detection offers groundbreaking solutions for securing and maintaining data integrity. A key benefit of blockchain is its unchangeable nature, which can be utilized for recording fraud evidence. Each transaction on the blockchain is time-stamped and permanently logged, making it virtually impossible to modify or delete. By developing blockchain-based frameworks for fraud detection, organizations can ensure a verifiable and tamper-proof record of all transaction activities. This not only enhances transparency but also simplifies audits and investigations, as stakeholders can trace back through the transaction history without the risk of data manipulation.

Federated Learning

Federated learning presents another promising avenue for enhancing fraud detection models while maintaining data privacy. Traditional machine learning approaches require aggregating sensitive data from multiple sources, such as banks or financial institutions, to train robust models. This can raise security concerns and hinder collaboration due to data-sharing regulations. Federated learning enables multiple organizations to jointly train models without exchanging raw data. Each organization trains a local model on its own data and shares only model updates (such as gradients) with a central server. This method not only safeguards user privacy but also improves the model's performance by leveraging a wide variety of data from different sources. Future research can explore techniques for improving communication efficiency, ensuring model generalization, and addressing challenges related to model divergence across institutions.

FUTURE ENHANCEMENTS

Explainable AI (XAI)

As the importance of transparency grows in the finance sector, future improvements to the AFDS will focus significantly on explainable AI (XAI). This approach aims to make complex ML models interpretable to end-users and stakeholders, thereby increasing trust and understanding. XAI will help demystify how the model arrives at specific decisions regarding fraud detection, allowing stakeholders to gain insight into the underlying factors contributing to alerts. This is especially important in regulated environments where transparency is essential for compliance and accountability.

Decentralized Fraud Intelligence Sharing

Another forward-looking enhancement to the AFDS involves the implementation of decentralized fraud intelligence sharing platforms. By leveraging blockchain technology or distributed ledgers, financial institutions can collaborate in sharing intelligence on fraudulent patterns and emerging threats without compromising sensitive customer information. This decentralized approach enables organizations to learn from one another's experiences, adapt to evolving threats more quickly, and collectively strengthen fraud detection capabilities industry-wide. Such collaboration can foster a community of shared knowledge, creating a safer financial ecosystem.

SIGNIFICANCE

Fraud detection within government operations is critical to maintaining public trust, safeguarding national resources, and ensuring the integrity of administrative processes. Conventional fraud detection approaches, which typically depend on manual reviews and rule-based systems, are becoming less effective against the advanced techniques employed by today's fraudsters. Machine learning introduces a transformative approach by enabling automated, real-time detection of anomalies across large and complex datasets. By analyzing patterns in past data, machine learning models can more accurately and efficiently predict and detect potential fraudulent activities.

Implementing machine learning in governmental fraud detection not only improves response times but also enhances transparency and accountability across departments. It minimizes financial losses, improves resource distribution, and aids in making proactive decisions. Moreover, machine learning systems can continuously adapt to evolving fraud patterns, making them far more resilient than static rule-based frameworks. Therefore, the adoption of machine learning for fraud detection represents a significant advancement in promoting ethical governance and protecting public interests.

The application of machine learning in detecting fraud within government systems holds substantial significance in enhancing operational efficiency and public sector accountability. Unlike manual methods, machine learning models can process massive amounts of financial transactions, procurement data, and citizen services records to uncover hidden patterns of fraud that might go unnoticed. Early detection of fraudulent activities minimizes monetary losses and reduces reputational damage to government institutions. Furthermore, integrating machine learning fosters a data-driven culture, where policies and enforcement strategies can be continuously refined based on real-time insights. This technological advancement ultimately strengthens governance frameworks and builds greater citizen confidence in public administration.

CONCLUSION

The Fraud Detection System (FDS) stands at the forefront of combating increasingly sophisticated fraudulent activities in the financial sector. By harnessing the power of cutting-edge machine learning techniques, it successfully integrates supervised, unsupervised, and rule-based methods to create a comprehensive approach that significantly enhances detection accuracy. This hybrid model not only identifies known fraud patterns but also adapts to new threats by leveraging data from various sources, offering a dynamic solution to a continually evolving problem.

The AFDS's capability for real-time processing ensures that potential fraud is identified and mitigated swiftly, minimizing financial losses and preserving customer trust. Moreover, its emphasis on operational efficiency results in substantial cost savings, automating the detection process and allowing institutions to focus resources on strategic initiatives rather than manual reviews.

Looking forward, two critical enhancements: explainable AI (XAI) and decentralized fraud intelligence sharing, promise to enhance the effectiveness and transparency of the AFDS. Adopting XAI will provide stakeholders with a clearer understanding of the model's decision-making process, promoting increased trust and ensuring adherence to regulatory standards. Concurrently, decentralized

intelligence sharing platforms will facilitate collaboration among financial institutions, enabling them to pool their knowledge of emerging fraud patterns while safeguarding sensitive data.

In summary, the evolution of the AFDS represents not only a technological advancement but also a paradigm shift in how organizations approach fraud detection. As it continues to evolve, the AFDS will play a crucial role in creating a safer and more resilient financial ecosystem, ultimately strengthening the integrity of financial transactions and enhancing consumer confidence in digital finance. The commitment to continuous improvement and collaboration will be vital as institutions navigate the complexities of fraud in an ever-changing landscape.

REFERENCES

1. Abdallah A, Maarof MA, Zainal A. Fraud detection system: A survey. J Netw Comput Appl. 2016 Jun 1; 68: 90–113.
2. Sadare K, Bhatt A, Tidake S. An Efficient Credit Card Fraud Detection Using SMOTE Under Machine Learning Environment. In International conference on soft computing for problem-solving. Singapore: Springer Nature Singapore; 2023 Aug 10; 625–634.
3. Van Vlasselaer V, Eliassi-Rad T, Akoglu L, Snoeck M, Baesens B. Gotcha! network-based fraud detection for social security fraud. Manag Sci. 2017 Sep; 63(9): 3090–110.
4. Phua C, Lee V, Smith K, Gayler R. A comprehensive survey of data mining-based fraud detection research. arXiv preprint arXiv:1009.6119. 2010 Sep 30.
5. West J, Bhattacharya M. Intelligent financial fraud detection: a comprehensive review. Comput Secur. 2016 Mar 1; 57: 47–66.
6. Abitova G, Abalkanov M. Comparative analysis of ML algorithms for fraud detection. 2024 IEEE 4th International Conference on Smart Information Systems and Technologies (SIST), Astana, Kazakhstan, 2024. p. 554–9. doi:10.1109/SIST61555.2024.10629283.
7. Whitrow C, Hand DJ, Juszczak P, Weston D, Adams NM. Transaction aggregation as a strategy for credit card fraud detection. Data Min Knowl Discov. 2009 Feb; 18: 30–55.
8. Bahnsen AC, Aouada D, Stojanovic A, Ottersten B. Feature engineering strategies for credit card fraud detection. Expert Syst Appl. 2016 Jun 1; 51: 134–42.
9. Fiore U, De Santis A, Perla F, Zanetti P, Palmieri F. Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. Inf Sci. 2019 Apr 1; 479: 448–55.
10. Sahin Y, Duman E. Detecting credit card fraud by ANN and logistic regression. In 2011 IEEE international symposium on innovations in intelligent systems and applications. 2011 Jun 15; 315–319.
11. Jurgovsky J, Granitzer M, Ziegler K, Calabretto S, Portier PE, He-Guelton L, Caelen O. Sequence classification for credit-card fraud detection. Expert Syst Appl. 2018 Jun 15; 100: 234–45.
12. Benchaji I, Douzi S, El Ouahidi B. Credit card fraud detection model based on LSTM recurrent neural networks. J Adv Inf Technol. 2021 May; 12(2): 113–118.
13. Rodrigues VF, Policarpo LM, da Silveira DE, da Rosa Righi R, da Costa CA, Barbosa JL, Antunes RS, Scorsatto R, Arcot T. Fraud detection and prevention in e-commerce: A systematic literature review. Electron Commer Res Appl. 2022 Nov 1; 56: 101207.
14. Laleh N, Abdollahi Azgomi M. A taxonomy of frauds and fraud detection techniques. In International Conference on Information Systems, Technology and Management. Berlin, Heidelberg: Springer Berlin Heidelberg; 2009 Mar 12; 256–267.
15. Óskarsdóttir M, Ahmed W, Antonio K, Baesens B, Dendievel R, Donas T, Reynkens T. Social network analytics for supervised fraud detection in insurance. Risk Anal. 2022 Aug; 42(8): 1872–90.

