University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR)

ISSN: 2714-3627

A Journal of the Department of Computer Science, University of Ibadan, Ibadan, Nigeria

Volume 14 No. 1, June, 2025

journals.ui.edu.ng/uijslictr
http://uijslictr.org.ng/

University of Ibadan
Journal of Science and Logics in ICT Research

Public Servant Service Watch System: Leveraging Artificial Intelligence, Machine Learning, and Big Data Analytics to Combat Corruption in Nigeria

[1]Adeniran, O. J. and [2]✉ Ojo, A. K.

[1,2]Department of Computer Science, University of Ibadan, Nigeria.

[1]adexjamez@gmail.com, [2]adebolak.ojo@gmail.com

Abstract

Public sector fraud continues to undermine governance and development efforts in Nigeria. Despite ongoing anti-corruption campaigns, existing detection mechanisms remain manual, reactive, and insufficiently equipped to flag complex financial irregularities in real time. A critical research gap exists in the integration of automated, data-driven approaches to proactively detect fraud among public officials. This study seeks to bridge that gap by developing and evaluating a machine learning–based system tailored for detecting suspicious financial behaviours using asset declarations and transaction records. The work employed two datasets: a synthetically generated dataset created with Python's Faker library and publicly available financial transaction data from Kaggle. These were harmonized using unique identifiers, cleaned, and pre-processed to support analysis. Exploratory Data Analysis (EDA) helped uncover patterns relevant to fraud detection, such as transaction spikes and discrepancies between income and declared assets. A Random Forest classifier was chosen for its balance of predictive performance and interpretability. The model was trained and deployed using Microsoft Azure to enable scalable, real-time processing. Results indicate that the Public Servant Service Watch system effectively identifies anomalies such as sudden asset accumulation and undeclared financial interests. The Random Forest model achieved high scores across accuracy, precision, recall, and AUC-ROC metrics. This study demonstrates the feasibility and impact of applying machine learning within a cloud-based infrastructure to improve transparency, accountability, and fraud prevention in the Nigerian public sector.

Keywords: *Fraud Detection, Public Sector Accountability, Machine Learning, Random Forest Classifier, Financial Forensics*

1. Introduction

Corruption remains a profound global challenge, affecting both developed and developing countries' governance, economic progress, and public trust. Particularly damaging is corruption by public officials, which often involves bribery, embezzlement, money laundering, illicit enrichment, and the abuse of power for private benefit. Such misconduct frequently entails the misuse of public funds and the creation of elaborate concealment networks involving shell companies, offshore accounts, and real estate investments [1] [2].

Estimates from the World Economic Forum and the United Nations suggest that corruption drains about 5% of global GDP annually, equivalent to trillions of dollars [1]. This financial haemorrhage severely undermines public institutions, reduces investment, and perpetuates social inequality, especially when high-ranking officials are involved [1, 3].

Digital fraud detection technologies—particularly machine learning—have emerged as powerful tools in addressing this problem. By analysing large volumes of financial transactions, machine learning models can uncover subtle patterns of fraudulent behaviour that traditional methods fail to detect. These systems improve over time by adapting to new corruption schemes and reducing false positives [4, 5, 6].

Nonetheless, the deployment of these intelligent systems faces significant obstacles.

Adeniran, O. J. and Ojo, A. K. (2025). Public Servant Service Watch System: Leveraging Artificial Intelligence, Machine Learning, and Big Data Analytics to Combat Corruption in Nigeria. *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR),* Vol. 14 No. 1, pp. 71 - 80

Access to high-quality data remains scarce: public official financial data often exists in fragmented, confidential, or incomplete forms. Moreover, consolidating diverse datasets from government agencies, banks, and other entities can be technically demanding and computationally expensive.

To address these constraints, this study combines publicly available data sources (e.g., Kaggle) with synthetic data generation techniques to simulate missing features such as declarant information. This hybrid dataset enhances model training and validation, leading to more robust and effective systems capable of detecting suspicious transactions.

2. Related Works

Several researchers have investigated the application of machine learning (ML) in detecting and preventing fraudulent financial transactions. Traditional approaches often relied on rule-based systems, which lacked adaptability and scalability. In contrast, machine learning models continuously learn from large datasets to identify patterns and anomalies, enabling real-time responses to emerging threats [7, 8, 9] proposed an AIS-based Fraud Detection Model (AFDM) inspired by the biological immune system to detect anomalies. Randhawa *et. al* [10] analysed various ML classifiers such as Naive Bayes, Random Forests, Decision Trees, and Support Vector Machines, and introduced a hybrid ensemble model using AdaBoost and majority voting to improve detection rates. Their study demonstrated that hybrid techniques are more resilient to noisy data and better at handling imbalanced datasets.

Preprocessing plays a vital role in fraud detection, as raw financial data often contain noise, outliers, and missing values that can reduce model accuracy. Smith and Patel [11] explored methods for consolidating data from Excel files, relational databases, and APIs using ETL pipelines and tools like Talend and Apache NiFi. Lee and Wong [12] found that median-based outlier capping and interpolation significantly enhanced classification performance. Feature engineering and selection further improve model accuracy and efficiency by identifying predictors such as transaction frequency and income-to-expense ratios. Gupta and Rao [13] recommended using mutual information and recursive feature elimination to retain only the top 20–30% of features, which helped maintain high recall while reducing complexity.

Model training and deployment have also been optimized through cloud-based platforms. Microsoft Azure's AutoML, for instance, uses Bayesian optimization and parallel processing to expedite hyperparameter tuning, achieving up to 60%-time savings compared to on-premises setups (Microsoft, 2023). Once trained, models are typically serialized using Python's pickle or joblib modules, which ensure efficient storage and reuse [14, 15]. Random Forest classifiers have consistently outperformed other models in fraud detection tasks, achieving high accuracy and AUC scores [16, 17].

While powerful, complex models can be difficult to interpret. Tools like SHAP (Shapley Additive Explanations) from the emerging field of Explainable AI (XAI) are now widely used to visualize and explain model outputs, making machine learning more transparent for decision-makers [18. 19].

3. Methodology

3.1 Overview of the Model Development Workflow

Figure 1 presents the structured workflow of the fraud detection model. The approach is based on the Cross-Industry Standard Process for Data Mining (CRISP-DM) framework and includes steps such as data acquisition, preprocessing, feature engineering, model training, evaluation, and deployment. The study uses a hybrid dataset comprising real and synthetic data to simulate complex financial patterns observed in public sector behaviour.

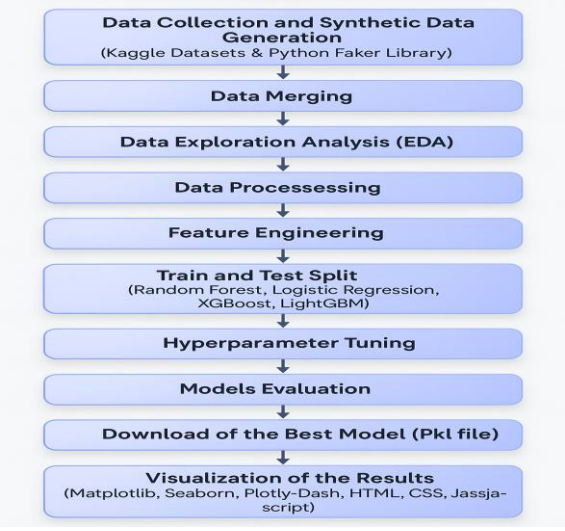

Figure 1: Structured Workflow of the Fraud Detection Model

3.2 Data Collection and Synthetic Data Generation

Due to limited access to official public servant financial records, two primary datasets were used:

- Kaggle Credit Card Transactions Dataset (20,000 labelled records)
- Synthetic Asset Declaration Dataset generated using the Python Faker library

The datasets were integrated using a synthetic unified identifier (e.g., User ID + pseudo BVN) to reflect realistic financial activity. This integration provided:

- Balanced class representation
- Improved model generalizability
- Simulated edge-case corruption patterns

Table 1 outlines the key features present in the Kaggle transaction dataset.

Table 1: Column Features for transaction dataset (Kaggle)

Column Name	Data Type
User	int64
Card	int64
Year	int64
Month	int64
Day	int64
Time	object
Amount	object
Use Chip	object
Merchant Name	int64
Merchant City	object
Merchant State	object
Zip	float64
MCC	int64
Errors?	object
Is Fraud?	object

3.3Addressing Class Imbalance with SMOTE

The initial class distribution was highly imbalanced (fraudulent: legitimate ≈ 1:100). This was corrected using SMOTE (Synthetic Minority Oversampling Technique), which interpolates new minority class samples between existing ones and their k-nearest neighbours [20]. This is shown in equation (1):

$$x_{new} = x_i + \delta.(x_{knn} - x_i), \delta \sim U(0,1) \quad (1)$$

Where x_i is a minority class instance, and x_{knn} is one of its k-nearest neighbors. This approach generates new, realistic examples that allow classifiers to better learn fraud patterns.

(a) Before SMOTE

Fraudulent cases: ~200 (1%) Legitimate cases: ~19,800 (99%) Result: The random forest tended to predict the proportion of legitimate cases in almost all samples, but it missed most fraud cases.

(b) After SMOTE

Fraudulent cases: ~19,800 (over-sampled by synthetic examples). Legitimate cases: ~19,800.

Benefits: A balanced 1:1 ratio forces the model to learn patterns related to fraud, improves the recall and F1 scores of the minority group.

3.4Data Exploration Analysis (EDA)

EDA techniques applied include:

- Histograms: To visualize transaction volume distribution
- Boxplots: For outlier detection
- Correlation matrices: To examine relationships between variables

Libraries used: Matplotlib, Seaborn, and Pandas.

3.5Data Preprocessing

Prioritization of the raw data to a form suitable for machine learning contributed to the misconception of a sacred bullet. The main steps included converting the field Amount from a string of currency symbols to a numeric type and parsing the separate fields Year, Month, Day, and Time into a single date object.

Other time-based attributes, such as transaction hours, day of the week and weekend, have also been extracted. Unemployment was imputed for missing values in the numeric columns, ensuring consistency across the dataset. This thorough pre-processing eliminated potential problems that could affect the reliability of the report.

3.6Feature Engineering

Additional features designed to capture financial anomalies included:

- Spending Ratios:

$$Expense - to - Income\ Ratio = \frac{Total\ Monthly\ Expenditure}{Declared\ Monthly\ Income} \quad (2)$$

- Rolling Statistics: Applied over 7-day and 30-day windows
- Frequency Features: Daily/weekly/monthly transaction counts categorized by merchant type

These features enhanced the model's ability to differentiate between normal and fraudulent behaviour.

3.7Train and Test Split

For a reliable evaluation of the performance of the model, the single dataset was divided into training and testing subsets using the 80-20 divide. Strict sampling was used to preserve the inherent imbalance of the false labelling and to ensure that both samples were representative of the real-world distribution. This step was necessary to create models that would generalise well to unobservable data.

3.8Selection of Algorithms

Different algorithms have been selected to solve the fraud detection problem: Random Forest, Logistic regression, XGBoost, and LightGBM. These models have been chosen based on their complementary strengths in ambiguity. Random Forest, a robust tree-based ensemble method, and XGBoost and LightGBM, both gradient boosting frameworks, are particularly good at capturing nonlinear patterns in unbalanced data. Unemployment was included as a baseline linear classifier.

3.9Hyperparameter Tuning (Azure Cloud Compute)

Due to the computational requirements of optimizing complex models, hyperparameter tuning has been done on Azure cloud resources. This cloud-based approach has enabled efficient network searches across multiple parameter combinations, which has greatly accelerated the process. Azure not only reduces the length of training sessions, but it also provides the scalability needed to process large data sets. Tuned parameters have resulted in models with increased predictive power and robustness.

3.10 Model Serialization

The selected random forest model has been serialized to a pickle (.pkl) file for easy deployment. This step guarantees reproducibility and allows the model to be seamlessly integrated into the production environment, where it can process new data in real-time.

3.11 Real-Time Deployment

After selecting the model, the Random Forest classifier was applied in the web application environment. The implementation framework supports the detection of fraud in real time and allows for the continuous monitoring of financial transactions of public officials. The system is designed to alert the authorities to suspicious activities, thus facilitating early intervention.

3.12 Evaluation Metrics

The performance of the trained fraud detection models was evaluated using standard classification metrics suitable for imbalanced datasets. These metrics provide insights into the model's ability to correctly identify both fraudulent and legitimate transactions.

Accuracy: The ratio of correctly predicted observations to the total observations.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

TP: True Positives (fraud correctly predicted as fraud)

TN: True Negatives (legit correctly predicted as legit)

FP: False Positives (legit incorrectly predicted as fraud)

FN: False Negatives (fraud incorrectly predicted as legit)

Precision: Measures the proportion of true positive predictions among all positive predictions.

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

Recall (Sensitivity or True Positive Rate): Measures the ability of the model to identify all relevant cases.

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

F1 Score: Harmonic mean of Precision and Recall. It provides a balance between Precision and Recall.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (6)$$

AUC-ROC (Area Under the Receiver Operating Characteristic Curve): Indicates the model's capability to distinguish between classes. A higher AUC indicates better model performance in distinguishing fraud from legitimate transactions.

4. Results

4.1 Model Performance

Figure 2 presents the model performance in real time, where transactions are tested for their legitimacy using combined features.

Figure 3 presents the result of the prediction, that is, the probability of fraud

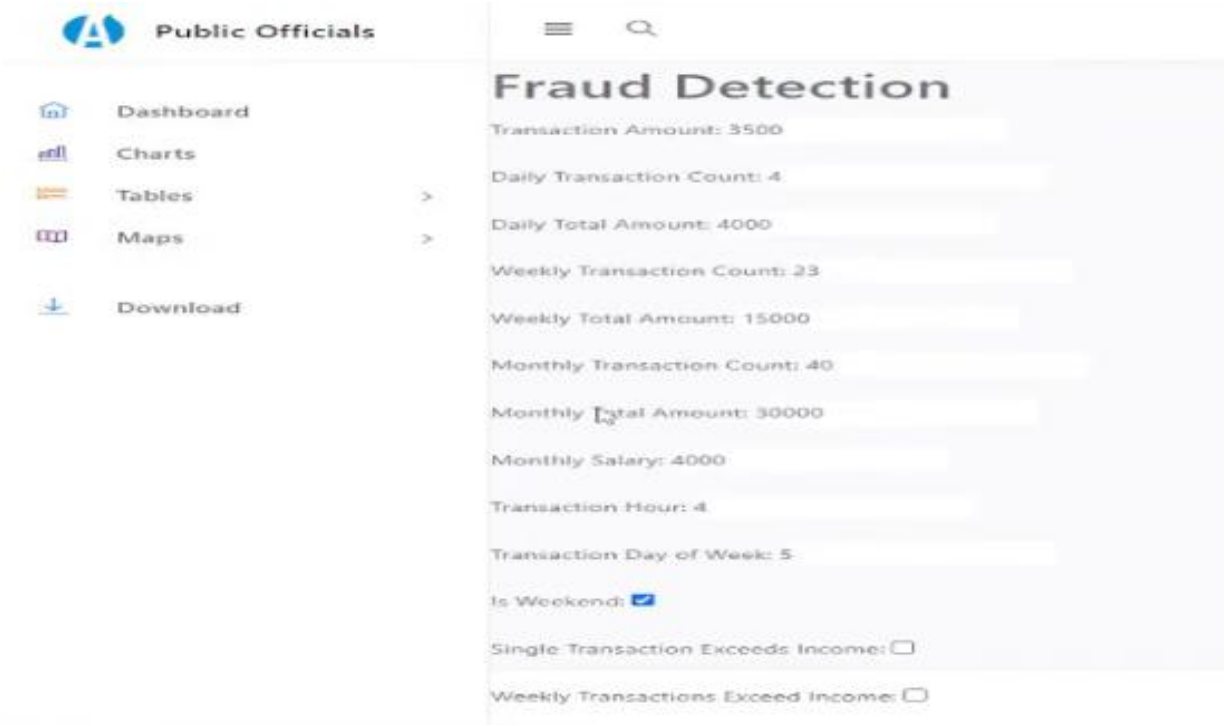

Figure 2: Real-time model performance evaluating transaction legitimacy based on combined features

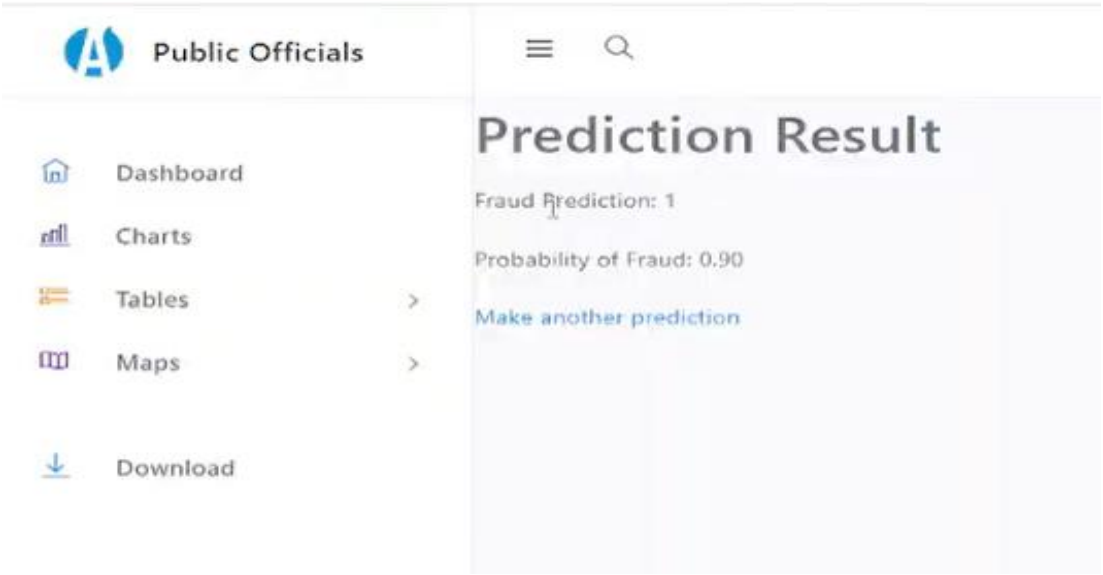

Figure 3: Probability of fraud

4.2 Model Evaluation and Comparison

The performance of four classification models—Logistic Regression, Random Forest, XGBoost, and LightGBM—was evaluated using standard metrics: accuracy, precision, recall, and F1-score for both fraudulent (Class 1) and legitimate (Class 0) transactions.

Logistic Regression achieved an overall accuracy of 91%, with excellent precision for Class 0 (1.00) but performed poorly on detecting fraud. Its precision and F1-score for Class 1 were only 0.01 and 0.02, respectively. These values reflect a strong bias toward the majority class, making it unsuitable for identifying irregular financial activity despite its high overall accuracy.

In contrast, Random Forest, XGBoost, and LightGBM all recorded perfect accuracy (1.00) across the board and showed consistent performance in detecting fraud. For Class 1, each of these ensemble models achieved a precision, recall, and F1-score of 0.80. Their performance remained perfect for Class 0, indicating their ability to balance detection across both classes. Notably, Random Forest achieved the highest overall precision (0.9967), slightly outperforming XGBoost and LightGBM (0.9780 each).

Table 2 presents the ROC curve comparison across models. The ensemble methods significantly outperformed Logistic Regression, offering a more reliable distinction between fraudulent and legitimate transactions. Logistic Regression, while accurate in general classification, lacked sensitivity to fraud cases, making it unreliable for real-world application in fraud detection.

4.3 Performance Evaluation for the four models

The confusion matrix is a central tool in evaluating the effectiveness of fraud detection models. It breaks down the model's predictions into four categories: true positives (correctly identified fraud), true negatives (correctly identified legitimate transactions), false positives (legitimate transactions wrongly flagged as fraud), and false negatives (fraudulent transactions missed by the model). This breakdown enables a clearer understanding of how each model handles the inherent class imbalance in fraud detection.

Figure 4 illustrates the confusion matrix for LightGBM, showing a balanced performance with a high number of correct classifications in both classes. It identifies fraud effectively while keeping false positives low.

Table 2: ROC curve comparison across models

Model	Accuracy	Precision (Class 1)	Recall (Class 1)	F1-Score	F1-Score (Class 1)	ROC-AUC	Overall Remarks
Random Forest	100%	0.80	0.80	1.00	0.80	0.9967	Strong detection of both classes, most balanced model
XGBoost	100%	0.80	0.80	1.00	0.80	0.9780	Reliable and consistent, effective for imbalanced data
LightGBM	100%	0.67	0.80	1.00	0.73	0.90	Fast and accurate, matches XGBoost in performance
Logistic Regression	91%	0.01	0.80	0.95	0.02	0.9037	High overall accuracy, but fails to detect fraud effectively

In contrast, Figure 5, representing Logistic Regression, reveals significant limitations. Although the model achieves reasonable accuracy overall, it fails to detect fraudulent transactions reliably misclassifying most fraud cases as legitimate, which severely impacts its usefulness in a real-world fraud detection setting. Figure 6 presents the confusion matrix for Random Forest, highlighting its strength in capturing both legitimate and fraudulent cases. It maintains high precision and recall, making it a dependable model for operational deployment.

Similarly, Figure 7 shows the performance of XGBoost, which performs comparably to Random Forest. It effectively minimizes misclassifications across both classes, demonstrating strong predictive capacity in imbalanced datasets.

These matrices highlight not only raw performance but also help in identifying which models are better suited for balancing risk—ensuring that fraud is detected without overwhelming systems with false alarms.

Figure 4: Confusion Matrix – LightGBM

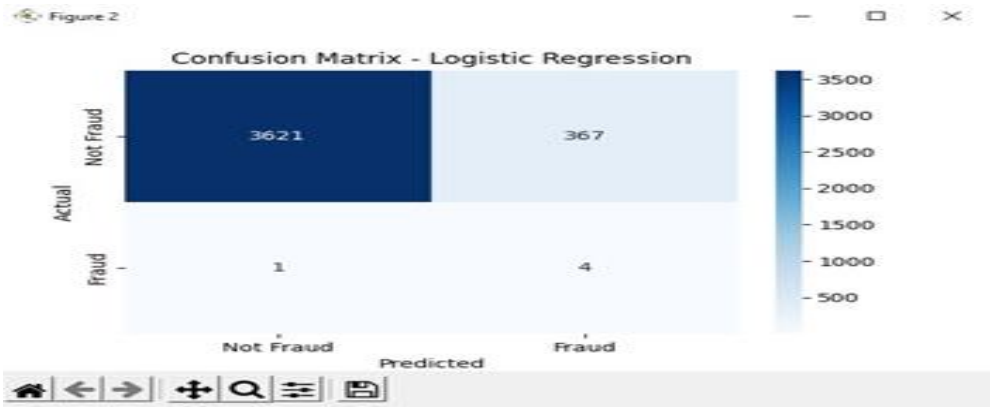

Figure 5: Confusion Matrix – Logistic Regression

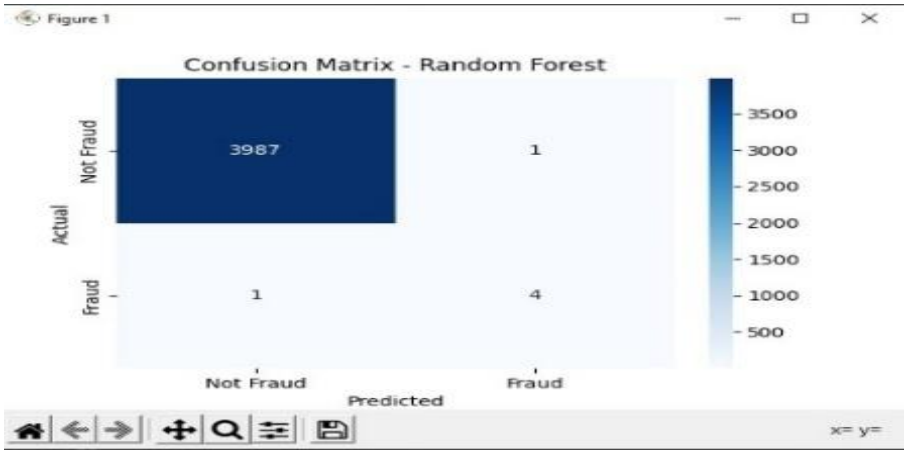

Figure 6: Confusion Matrix – Randome Forest

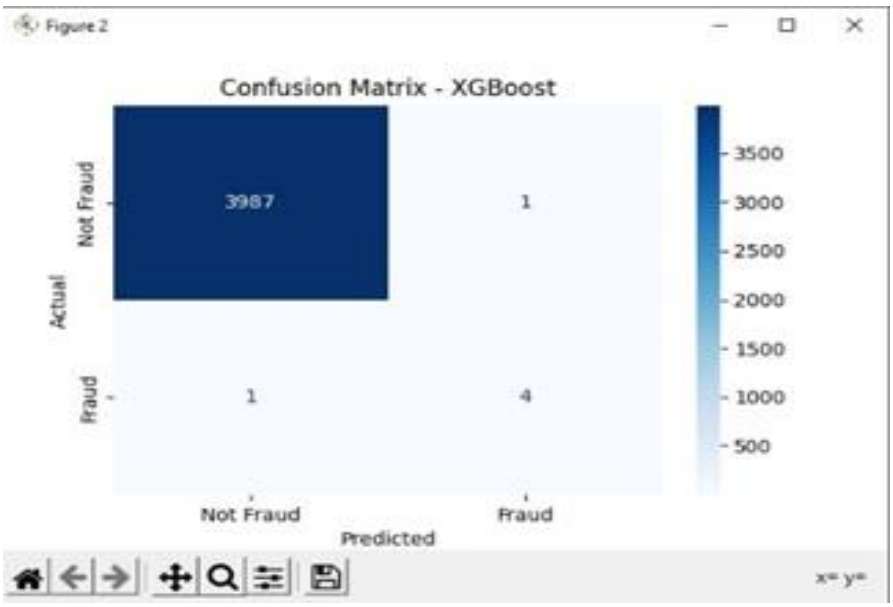

Figure 7: Confusion Matrix – XGBoost

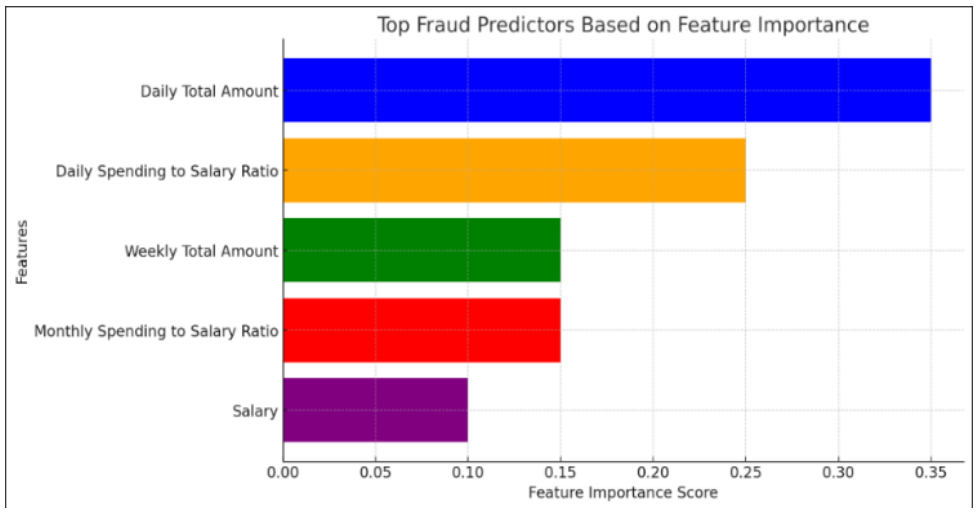

Figure 8: Feature importance scores influencing the model's decision-making

4.4 Model Explainability

Model explainability is crucial in fraud detection as it builds trust and transparency, allowing stakeholders to understand why a transaction is flagged. It ensures compliance with legal standards, supports human investigators in verifying alerts, and helps reduce false positives by revealing key patterns and feature influences. Feature importance scores highlight which variables most influence the model's decisions, offering insights into behavioural red flags such as unusually large transactions or frequent foreign transfers. This not only aids in debugging and improving model accuracy but also strengthens accountability and auditability in high-stakes environments like finance and public service. By clearly showing which features drive predictions, organizations can better justify automated decisions and make more informed policy or investigative responses.

Figure 12: This presents the feature importance score for the features that influence the model's decision making.

4.5 Visualization of the Results

To support interpretation and stakeholder engagement, the results were visualized using a range of tools including Matplotlib, Seaborn, and Plotly Dash. Histograms were used to analyse the distribution of transaction amounts and frequency, while box plots helped highlight potential outliers that might signify fraudulent

behaviour. Correlation heatmaps exposed relationships among input features, guiding feature selection and model refinement.

The use of interactive dashboards, built with HTML, CSS, and JavaScript, provided a dynamic interface for monitoring fraud detection metrics in real time. These visual elements enabled stakeholders to explore trends, track model outputs, and interpret findings without needing technical expertise, thereby enhancing transparency and usability.

4.6 Justification for Model Selection

Random Forest was chosen as the preferred model due to its strong balance between performance, speed, and interpretability. It aggregates multiple decision trees, providing high precision and recall without overfitting. Its resilience to missing or noisy data makes it particularly suited to real-world financial environments, where data irregularities are common.

Furthermore, Random Forest offers built-in feature importance analysis, allowing analysts to identify which variables—such as transaction patterns or asset declarations—most influence the model's output. This transparency enhances both trust in the system and the ability to conduct follow-up investigations.

While XGBoost and LightGBM also demonstrated excellent predictive power, Random Forest offered a shorter training time and simpler tuning, making it more practical for iterative deployment and updates. Its robustness and ease of interpretation make it a strong candidate for scalable and transparent fraud detection in financial systems.

5. Conclusion

This study investigated the potential of machine learning in detecting financial irregularities among public officials. By integrating both real-world and synthetically generated financial data, multiple classification models were developed and evaluated. Ensemble-based approaches such as Random Forest and XGBoost demonstrated superior performance in distinguishing between legitimate and suspicious transactions. Their accuracy was enhanced through the creation of meaningful features, including salary-to-spending ratios and rolling statistical metrics, which provided critical behavioural insights.

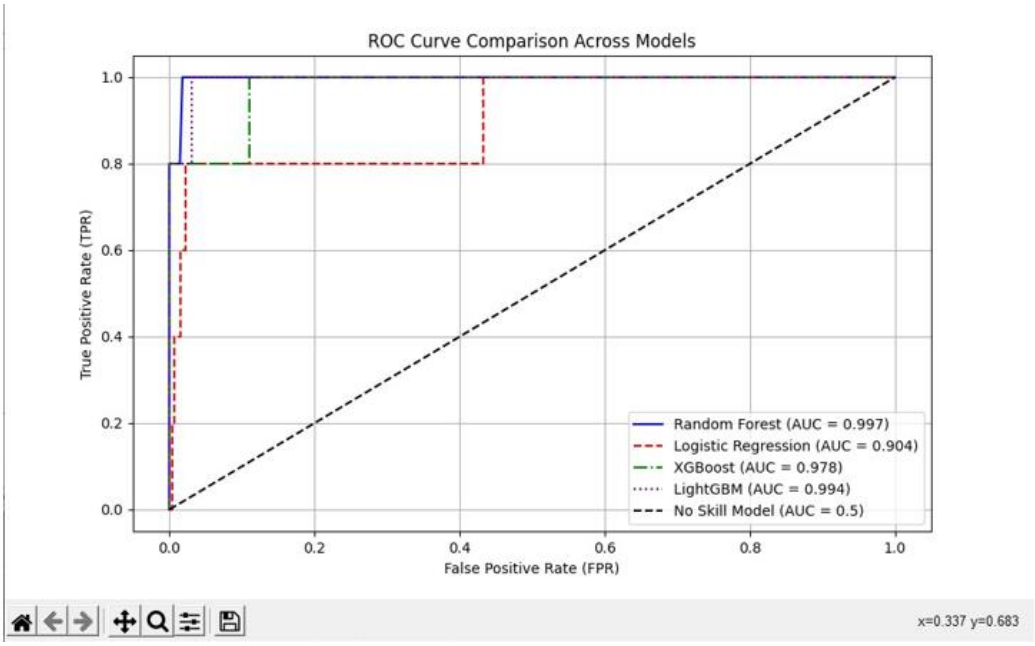

Figure 9: ROC curve comparison across models

The use of cloud infrastructure, particularly Microsoft Azure, enabled efficient model training and scalability, making the system suitable for large-scale implementation. Nonetheless, the study faced limitations. It focused exclusively on digital transactions, omitting cash-based activities that often play a role in corruption cases. Furthermore, the relatively small number of confirmed fraud instances within the dataset presented challenges for generalizability, although preserving the natural class imbalance was important for realism.

Model interpretability emerged as a key concern. While the chosen models performed well, their complexity may hinder adoption in regulatory environments that demand transparency and explainability. Incorporating explainable AI techniques, such as SHAP (Shapley Additive Explanations), is recommended in future developments to enhance trust and usability.

In conclusion, this research highlights the viability of applying machine learning techniques to support fraud detection and accountability efforts within the public sector. Future work should focus on expanding the dataset, incorporating additional transaction types, and improving model transparency to ensure practical deployment and impact.

References

[1] U. Nations, "Global cost of corruption at least 5 per cent of world gross domestic product, Secretary-General tells Security Council, citing World Economic Forum data," 10 September 2018. [Online]. Available: https://press.un.org/en/2018/sc13493.doc.htm.

[2] W. Bank, "The cost of corruption: Global economic impacts," World Bank, Washington, D.C. , 2022.

[3] United Nations & World Bank, "Finding fraud: Early detection of fraud and corruption in public procurement through technology," World Bank, Washington, D.C., 2021.

[4] M. K. H. Chy, "Proactive fraud defense: Machine learning's evolving role in protecting against online fraud," 2024.

[5] Chen, Y.; Zhao, C.; Xu, Y.; Nie, C., "Year-over-year developments in financial fraud detection via deep learning: A systematic literature review," arXiv, 2025.

[6] Mastercard, "Mastercard's AI-powered fraud detection system, Decision Intelligence, processes billions of transactions," Business Insider, 2025.

[7] Bouchama, F.; Kamal, M., "Enhancing cyber threat detection through machine learning-based behavioral modeling of network traffic patterns," *International Journal of Business Intelligence and Big Data Analytics*, vol. 4, no. 9, p. 1–9, 3 September 2021.

[8] Bagaa, M.; Taleb, T.; Bernabe, J. B.; Skarmeta, A., "A machine learning security framework for IoT systems," *IEEE Access*, vol. 8, p. 114066–114077, 2020.

[9] V. Shah, "Machine learning algorithms for cybersecurity: Detecting and preventing threats," *Revista Española de Documentación Científica*, vol. 15, no. 4, p. 42–66, 2021.

[10] Randhawa, K.; Loo, Chu Kiong; Seera, M.; Lim, C. P.; Nandi, A. K., "Credit card fraud detection using AdaBoost and majority voting," *IEEE Access*, vol. 6, p. 14277–14284, 2018.

[11] Smith, J.; Patel, R., "Data integration from heterogeneous sources: A review," *Journal of Data Engineering*, vol. 10, no. 1, p. 23–35, 2019.

[12] Lee, K.; Wong, T., "Preprocessing techniques for anomaly detection in financial data," *Journal of Financial Analytics*, vol. 5, no. 4, p. 45–60, 2018.

[13] Gupta, S.; Rao, P., "Feature engineering and selection methods for fraud detection," *International Journal of Machine Learning*, vol. 14, no. 2, p. 100–115, 2020.

[14] J. Brownlee, "Save and load machine learning models in Python with scikit-learn," 2019. [Online]. Available: https://machinelearningmastery.com/save-load-machine-learning-models-python-scikit-learn/.

[15] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; ...;

Duchesnay, É., “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research,* vol. 12, p. 2825–2830, 2011.

[16] Johnson, M.; Desai, A., “Random Forest performance in detecting fraudulent transactions,” *Journal of Fraud Analytics,* vol. 3, no. 2, p. 67–82, 2021.

[17] Afriyie, J. K.; Tawiah, K.; Pels, W. A.; Addai-Henne, S.; Dwamena, H. A.; Owiredu, E. O.; Ayeh, S. A.; Eshun, J., “A supervised machine learning algorithm for detecting and predicting fraud in credit card transactions,” *Decision Analytics Journal,* vol. 6, no. 100163, 2023.

[18] Faruk, U.; Tariq, M.; Oladele, B. A.; Gok, S., “ Explainable AI for fraud detection: Building trust and transparency in AI-driven financial security systems,” *Decision Analytics Journal,* vol. 7, no. 100176, 2025.

[19] Lin, K.; Gao, Y., “Model interpretability of financial fraud detection by group SHAP,” *Expert Systems with Applications,* vol. 210, no. 118354, 2022.

[20] Chawla, Nitesh V.; Bowyer, Kevin W.; Hall, Lawrence O.; Kegelmeyer, W. Philip, “SMOTE: Synthetic Minority Over-sampling Technique,” *Journal of Artificial Intelligence Research,* vol. 16, p. 321–357, 2002.