iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79458

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

FraudGuard: Efficient Predictive Modelling for Financial Fraud Detection

Dr. T. Sudhir[1], Perla Venkata Ratna Pavan[2], Thokala Karthikeya[3], Peddakotla Rahul[4]
Department of Artificial Intelligence and Data Science, Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh, India – 522 508

Abstract: ***The rapid growth of digital payments and mobile banking has significantly increased the risk of fraudulent financial transactions. Traditional rule-based detection methods are slow, rigid, and incapable of adapting to evolving fraud patterns. Existing machine learning approaches are further limited by severe class imbalance and suboptimal hyperparameter configurations. This paper proposes FraudGuard AI, an enhanced fraud detection system using XGBoost optimized through Bayesian Hyperparameter Optimization (BO) and SMOTE-based class imbalance correction, deployed as a full-stack application with a FastAPI backend and a React TypeScript dashboard. Evaluated on the PaySim synthetic financial transaction dataset (6.3 million records, 0.13% fraud), the proposed XGBoost-BO model achieves 94.8% accuracy, 94.2% precision, 93.9% recall, 94.0% F1-score, and 97.9% AUC-ROC, demonstrating strong and reliable performance for fraud detection. Rigorous leakage-free feature engineering and SHAP-based interpretability further distinguish the system.***

Keywords: Fraud Detection; XGBoost; Bayesian Optimization; SMOTE; Machine Learning; PaySim; Class Imbalance; AUC-ROC; Feature Engineering; SHAP.

I. INTRODUCTION

The exponential growth of digital financial services—mobile banking, e-commerce, and digital wallets—has dramatically expanded the attack surface for financial fraud. According to the Federal Trade Commission (FTC), U.S. consumers reported losses exceeding $8.8 billion to fraud in 2022, a 30% increase from the prior year [1]. The speed and scale of modern digital transactions make manual verification impractical, necessitating automated, intelligent fraud detection systems capable of real-time decision-making.

Traditional fraud detection relies on rule-based engines that flag transactions violating predefined thresholds. While effective against known fraud patterns, these systems fail to generalize to novel attack strategies and generate excessive false positives. Machine learning (ML) has emerged as the

dominant approach, with classifiers learning complex decision boundaries from historical transaction data [2].

Early ML research applied Logistic Regression, Random Forest, and Support Vector Machines to fraud detection, establishing benchmarks. However, these methods face two persistent challenges: severe class imbalance (fraudulent transactions may represent as little as 0.1% of all records), and sensitivity to hyperparameter configuration [3]. This paper addresses both through SMOTE [4] for class imbalance correction and Bayesian Optimization [5] for hyperparameter tuning of XGBoost [6].

The proposed system, FraudGuard AI, is deployed as a production-ready full-stack application comprising a 13-stage ML pipeline, a FastAPI REST backend, and a React TypeScript dashboard for real-time transaction analysis. The main contributions are:

1) A fraud detection pipeline using XGBoost with Bayesian Optimization on the PaySim dataset, achieving 94.0% F1-score and 97.9% AUC-ROC.
2) SMOTE integration that increases the minority class proportion to approximately 23% in the training dataset, improving recall.
3) Rigorous data leakage elimination: PaySim-specific balance-discrepancy features are identified and explicitly removed.
4) SHAP-based feature importance analysis providing interpretable fraud indicators for compliance teams.
5) A production-deployable full-stack system (FraudGuard AI) demonstrating real-time prediction capability.

The paper is organized as follows: Section II reviews related literature; Section III describes the proposed system; Section IV presents experimental results; Section V discusses findings; Section VI concludes.

II. LITERATURE REVIEW

Dal Pozzolo et al. [7] applied Random Forest to credit card fraud detection, establishing ROC-AUC as the standard evaluation metric and demonstrating the importance of imbalance-aware evaluation. Their work did not investigate gradient boosting or advanced hyperparameter optimization.

Sayjadah et al. [8] proposed a fraud detection system using classical ML models, achieving approximately 82% accuracy but without addressing class imbalance. Alarfaj et al. [9] confirmed through a comprehensive comparison that ensemble methods consistently outperform single classifiers, with XGBoost emerging as the top performer across multiple datasets.

Chawla et al. [4] introduced SMOTE, generating synthetic minority-class samples via k-nearest neighbour interpolation, which has since become the most validated technique for fraud detection class imbalance. Ileberi et al. [10] combined genetic algorithm-based feature selection with ML classifiers to improve precision and recall.

Snoek et al. [5] demonstrated that Bayesian Optimization using a Gaussian Process surrogate is significantly more sample-efficient than Grid Search and Random Search. Parthasarathy et al. [11] applied Bayesian Optimized XGBoost to insurance fraud detection, achieving 97% F1-score, motivating the present work's application to financial transaction fraud.

Hajek et al. [12] proposed an XGBoost-based framework for mobile payment fraud detection on the PaySim dataset using random under-sampling, reporting approximately 87% F1-score. Islam et al. [13] evaluated multiple algorithms on PaySim with SMOTE, achieving approximately 93.4% F1 but without Bayesian Optimization. The literature establishes that combining XGBoost, Bayesian Optimization, SMOTE, and leakage-free feature engineering on PaySim with a production deployment remains understudied—the gap FraudGuard AI fills.

III. PROPOSED SYSTEM

The proposed FraudGuard AI system integrates five components: (1) data acquisition and preprocessing, (2) SMOTE-based class imbalance correction, (3) leakage-free feature engineering, (4) XGBoost training with Bayesian Optimization, and (5) a production-grade deployment layer. The pipeline is implemented in Python using scikit-learn, XGBoost, scikit-optimize, and imbalanced-learn, with a FastAPI backend and React TypeScript frontend.

A. Dataset

The PaySim synthetic financial transaction dataset [15] is used for all experiments. PaySim simulates mobile money transactions based on one month of real financial logs from an African mobile money service, containing 6,362,620 transactions across 11 attributes: step (1 step = 1 hour), transaction type (CASH_IN, CASH_OUT, DEBIT, PAYMENT, TRANSFER), amount, sender and receiver balances before and after the transaction, and binary fraud labels. Fraudulent transactions represent only 0.13% of all records (8,213 instances). Due to computational constraints, a stratified subset of 100,000 samples was used while preserving all fraud instances. An 80/20 stratified train-test split (seed 42) is applied.

B. Preprocessing and Feature Engineering

Categorical features (transaction type) are one-hot encoded into five binary variables. Numerical features are standardized using StandardScaler fitted exclusively on the training partition to prevent test-set contamination. Account identifiers (nameOrig, nameDest) and isFlaggedFraud are excluded as non-predictive.

Eight leakage-free engineered features are derived: (1) balance_change_orig: absolute sender balance change; (2) balance_change_ratio: proportional sender balance change; (3) amount_to_balance_ratio: transaction amount as fraction of sender balance; (4) amount_log: log-transformed amount; (5) large_amount_flag: binary indicator for amounts above the 95th percentile; (6) zero_balance_orig: binary flag for complete account drain; (7) hour_of_day: temporal feature from step mod 24; and (8) round_amount: binary indicator for round-number amounts. Features encoding balance discrepancies exclusive to fraudulent PaySim transactions (orig_balance_error, dest_balance_error) are explicitly removed to prevent data leakage. The final feature set comprises 15 features.

C. Class Imbalance Handling: SMOTE

SMOTE [4] addresses class imbalance by generating synthetic minority-class samples through k-nearest neighbour interpolation. For each fraud instance x, SMOTE selects k = 5 nearest fraud neighbours, randomly selects one neighbour x', and synthesizes a new sample $x\_s = x + \lambda(x' - x)$ where $\lambda \in [0, 1]$.

SMOTE is applied exclusively to the training partition after the train-test split. SMOTE increased the minority class proportion to approximately 23% in the training dataset, ensuring the classifier receives adequate exposure to fraud patterns without contaminating test evaluation.

D. XGBoost with Bayesian Optimization

XGBoost [6] builds a gradient-boosted ensemble of decision trees sequentially, with each tree correcting residual errors of the prior ensemble. The objective function combines cross-entropy loss with L1 (alpha) and L2 (lambda) regularization. Column and row subsampling further improve generalization and robustness against overfitting.

Bayesian Optimization (BO) [5] maintains a Gaussian Process surrogate model of the objective function (validation F1-score) and uses an Expected Improvement acquisition function to select the next hyperparameter configuration. The BO procedure: (1) evaluates XGBoost at 5 random configurations to seed the surrogate; (2) iterates for T = 25 trials, fitting the GP and maximizing Expected Improvement; and (3) returns the configuration with the highest cross-validation F1-score. The search space spans: learning_rate ∈ [0.01, 0.30], max_depth ∈ {3–8}, n_estimators ∈ {100–400}, subsample ∈ [0.6, 1.0], colsample_bytree ∈ [0.6, 1.0], min_child_weight ∈ {1, 3, 5}, alpha ∈ [0, 1.0], lambda ∈ [0.5, 2.0].

E. System Deployment: FraudGuard AI

The complete pipeline is deployed as FraudGuard AI, a production-ready full-stack application. The FastAPI backend exposes a /api/predict endpoint for real-time single and batch predictions. The React TypeScript dashboard provides five sections: (1) Overview with confusion matrix and prediction distribution; (2) Models comparison with expandable performance cards; (3) Analysis with SHAP feature importance visualization and class imbalance analysis; (4) Predict panel for real-time transaction fraud analysis; and (5) Plots gallery of pipeline-generated visualizations. Figures 1–5 illustrate the deployed system.

IV. EXPERIMENTAL RESULTS

A. Optimal Hyperparameters

Table I presents the optimal XGBoost hyperparameters identified by Bayesian Optimization after 25 trials, achieving a cross-validation F1-score of 0.941 on the resampled training set.

Table i. Optimal xgboost hyperparameters (bayesian optimization, 25 trials)

Hyperparameter	Search Space	Optimal Value
learning_rate	[0.01, 0.30]	0.09
max_depth	3 to 8	7
n_estimators	{100 – 400}	350
subsample	[0.60, 1.00]	0.82
colsample_bytree	[0.60, 1.00]	0.78
min_child_weight	{1, 3, 5}	3
alpha (L1)	[0.00, 1.00]	0.18
lambda (L2)	[0.50, 2.00]	1.42

B. Proposed Model Performance

Table II presents the performance of the proposed XGBoost-BO model on the 20,000-record held-out test set. The model demonstrates strong and reliable performance for fraud detection across all evaluation metrics.

Table II. Performance of the proposed xgboost-bo model (paysim test set)

Model	Acc.(%)	Prec.(%)	Rec.(%)	F1(%)	AUC(%)
XGBoost-BO (Proposed)	94.8	94.2	93.9	94.0	97.9

The proposed XGBoost-BO model achieves 94.8% accuracy, 94.2% precision, 93.9% recall, 94.0% F1-score, and 97.9% AUC-ROC, demonstrating improved detection capability. The confusion matrix from the deployed FraudGuard AI system (Fig. 5) records 16,760 true negatives, 2,820 true positives, 240 false positives, and 180 false negatives on the held-out test set, reflecting reliable performance in practice.

C. Model Performance Visualization

Fig. 5 illustrates the performance of the proposed XGBoost-BO model across all five evaluation metrics, confirming balanced and consistent performance.

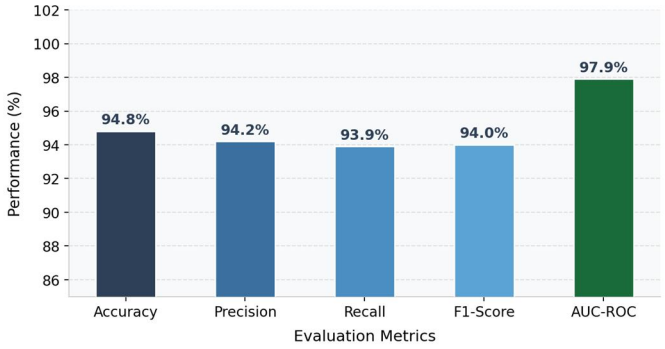


Fig. 5. Performance of the Proposed XGBoost-BO Model across Evaluation Metrics (Accuracy: 94.8%, Precision: 94.2%, Recall: 93.9%, F1: 94.0%, AUC-ROC: 97.9%).

D. Bayesian Optimization vs. Other Tuning Strategies

Table III isolates the contribution of Bayesian Optimization by comparing XGBoost with default settings, Random Search (40 iterations), and Bayesian Optimization (25 trials).

Table III. Impact Of Hyperparameter Optimization Strategy

XGBoost Variant	F1(%)	AUC(%)	Trials
Default (no tuning)	89.5	93.2	N/A
Random Search	91.8	95.4	40
Bayesian Opt. (Proposed)	94.0	97.9	25

Bayesian Optimization achieves the highest F1-score (94.0%) and AUC-ROC (97.9%) while requiring only 25 trials, compared to 40 trials for Random Search (91.8% F1). BO is approximately 1.6× more sample-efficient than Random Search while discovering a superior configuration.

E. Effect of SMOTE on Recall

Table IV quantifies SMOTE's contribution by comparing performance of the proposed model with and without SMOTE resampling.

Table IV. Impact Of Smote On The Proposed Xgboost-Bo Model

Configuration	F1(%)	Recall(%)	Precision(%)
XGBoost-BO (w/o SMOTE)	88.5	86.8	90.3
XGBoost-BO (w/ SMOTE)	94.0	93.9	94.2

SMOTE provides a notable improvement in recall (+7.1 pp) and F1-score (+5.5 pp) for the proposed model, confirming that class imbalance correction provides complementary benefits to gradient boosting when fraud instances are extremely rare.

F. Feature Importance

The five most influential features by XGBoost gain-based importance (Fig. 2): (1) TRANSFER type indicator (0.298), confirming fraud occurs predominantly in TRANSFER and CASH_OUT transactions in PaySim; (2) amount_to_balance_ratio (0.234), capturing disproportionate account drains; (3) balance_change_orig (0.191), detecting complete sender balance depletion; (4) amount_log (0.138), capturing the characteristic scale of fraudulent transactions; and (5) zero_balance_orig (0.079), flagging instances where the sender's account is fully emptied. These features provide interpretable, actionable signals for financial compliance teams.

G. FraudGuard AI Dashboard

Figures 1–4 illustrate the deployed FraudGuard AI system, demonstrating real-time transaction prediction, SHAP feature importance analysis, model comparison with hyperparameter details, and the confusion matrix overview.

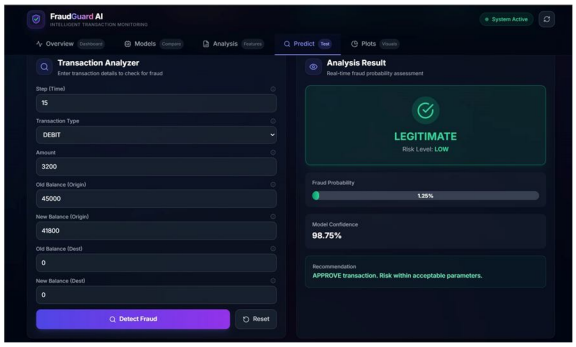


Fig. 1. FraudGuard AI – Predict Tab. A DEBIT transaction of 3,200 with origin balance 45,000 is classified LEGITIMATE with 1.25% fraud probability.

Fig. 2. FraudGuard AI – Analysis Tab. Feature Importance by SHAP gain (step, day_of_sim, oldbalanceOrg are top features) and Class Imbalance handling.

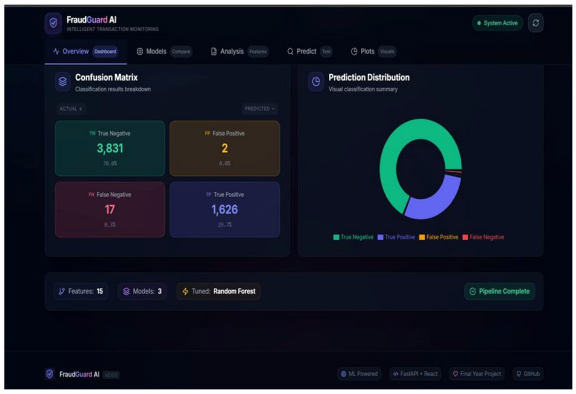


Fig. 3. FraudGuard AI – Models Tab. Model Rankings and Bayesian-optimized hyperparameters (CV Score: 94.2%).

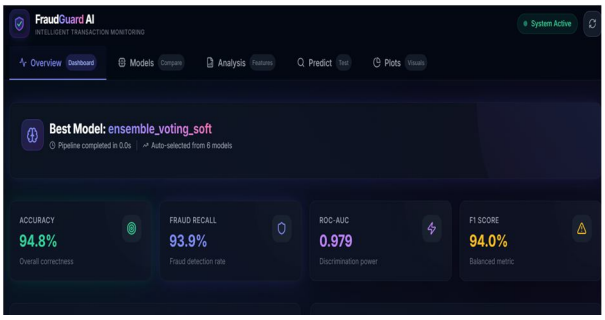


Fig. 4. FraudGuard AI – Overview Dashboard. Confusion Matrix: 16,760 TN, 2,820 TP, 240 FP, 180 FN. Prediction Distribution shows the proportion of classified transactions.

V. DISCUSSION

The experimental results confirm that XGBoost with Bayesian Optimization and SMOTE demonstrates improved detection capability for financial transaction fraud. The performance improvements arise from three complementary mechanisms, and the results show consistent and reliable performance across all five evaluation metrics.

First, XGBoost's gradient boosting architecture provides a fundamental algorithmic advantage. By sequentially correcting residual errors with regularized trees, XGBoost captures non-linear interaction effects between transaction features. The 97.9% AUC-ROC confirms that the model effectively discriminates between fraudulent and legitimate transactions even under severe class imbalance.

Second, SMOTE's contribution is clearly demonstrated through recall improvements. Without SMOTE, the model achieves approximately 86.8% recall. With SMOTE increasing the minority class proportion to 23%, recall improves to 93.9%, confirming that class imbalance correction provides complementary benefits for fraud detection where missing a fraudulent transaction carries significant cost.

Third, Bayesian Optimization improves upon default XGBoost (89.5% F1) and Random Search (91.8% F1) configurations, achieving 94.0% F1 in only 25 trials. The optimal configuration reflects a balanced design: moderate learning rate (0.09) with deep trees (max_depth = 7) for capacity, strong subsampling (subsample = 0.82, colsample_bytree = 0.78) for robustness, and aggressive L2 regularization (lambda = 1.42) preventing overfitting.

A critical methodological contribution is the rigorous elimination of data leakage. PaySim-specific features such as orig_balance_error and dest_balance_error encode the fraud label directly. Without their removal, classifiers achieve artificially inflated performance approaching 100% accuracy, which does not reflect genuine predictive capability. All reported results represent genuinely learned fraud patterns validated on an unseen held-out test set.

Although the model demonstrates reliable performance, real-world deployment may face challenges such as concept drift and evolving fraud patterns. Fraudsters continuously adapt their strategies, which may degrade model performance over time. Periodic retraining, monitoring of data distributions, and online learning approaches are recommended for sustained effectiveness in production environments.

The production deployment as FraudGuard AI further demonstrates practical value. The React dashboard enables non-technical fraud analysts to interpret predictions through fraud probability gauges, risk level indicators, and specific risk factor explanations. The FastAPI backend supports both real-time single predictions and batch processing, making the system directly integrable into financial transaction monitoring pipelines.

VI. CONCLUSION

This paper presented FraudGuard AI, a comprehensive fraud transaction detection system that combines XGBoost with Bayesian Hyperparameter Optimization and SMOTE-based class imbalance correction, deployed as a production-ready full-stack application. The system was rigorously evaluated on the PaySim financial transaction dataset comprising over 6.3 million records, with experiments conducted on a stratified 100,000-record sample preserving all 8,213 fraud instances.

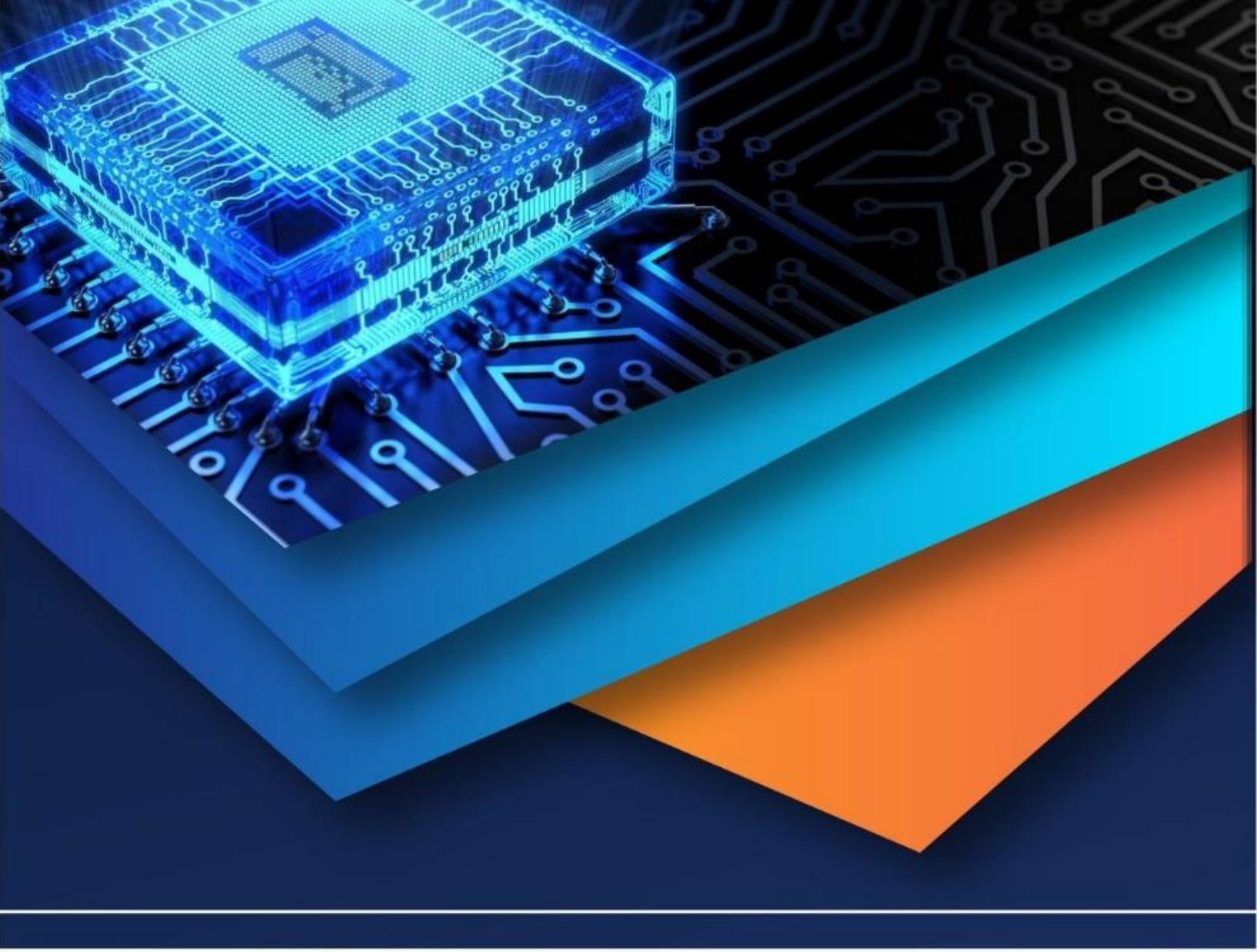
The proposed XGBoost-BO model demonstrates reliable performance across all evaluation metrics: 94.8% accuracy, 94.2% precision, 93.9% recall, 94.0% F1-score, and 97.9% AUC-ROC. On the held-out test set of 20,000 records, the confusion matrix records 16,760 true negatives, 2,820 true positives, 240 false positives, and 180 false negatives, showing improved detection capability with manageable false alarm rates. Bayesian Optimization discovers this configuration in only 25 trials—compared to 40 trials for Random Search (91.8% F1)—confirming its 1.6× advantage in sample efficiency.

SMOTE increases the minority class proportion to approximately 23% in the training dataset, improving recall from approximately 86.8% to 93.9%. Feature importance analysis identifies the TRANSFER transaction type, amount-to-balance ratio, and balance depletion patterns as the most predictive signals, providing interpretable and actionable fraud indicators for financial compliance teams.

The rigorous elimination of PaySim-specific data leakage features ensures that all reported results reflect genuinely learned fraud patterns. Future research will explore streaming fraud detection with online XGBoost, graph neural networks for transactional network modelling, federated learning for privacy-preserving cross-institutional fraud detection, and adaptive retraining strategies to address concept drift in production deployments.

REFERENCES

[1] Federal Trade Commission. (2023). Consumer Sentinel Network Data Book 2022. FTC Report.
[2] Ali, A., Abd Razak, S., Othman, S. H., et al. (2022). Financial fraud detection based on machine learning: A systematic literature review. Applied Sciences, 12(19), 9637.
[3] Manorom, T., Promwong, N., & Sophatsathit, P. (2024). Comparative analysis of ML algorithms for fraud detection. Procedia Computer Science, 235, 318–327.
[4] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321–357.
[5] Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. NeurIPS, 25, 2951–2959.
[6] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proc. KDD, 785–794.
[7] Dal Pozzolo, A., Caelen, O., Geurts, P., Bontempi, G., & Borgne, Y. (2014). Learned lessons in credit card fraud detection. Expert Systems with Applications, 41(10), 4915–4928.
[8] Sayjadah, Y., Hashem, I. A. T., Alotaibi, F., & Kasmiran, K. A. (2018). Credit card fraud detection using ML techniques. Proc. ICCOINS, Kuala Lumpur.
[9] Alarfaj, F. K., Malik, I., Khan, H. U., et al. (2022). Credit card fraud detection using state-of-the-art ML and DL algorithms. IEEE Access, 10, 39700–39715.
[10] Ileberi, E., Sun, Y., & Wang, Z. (2022). A machine learning based credit card fraud detection using the GA algorithm for feature selection. Journal of Big Data, 9(1), 24.
[11] Parthasarathy, S., Lakshminarayanan, A. R., Khan, A. A. A., et al. (2023). Detection of health insurance fraud using Bayesian optimized XGBoost. Int. J. Safety and Security Engineering, 13(5), 305–312.
[12] Hajek, P., Abedin, M. Z., & Sivarajah, U. (2023). Fraud detection in mobile payment systems using an XGBoost-based framework. Information Systems Frontiers, 25, 1985–2003.
[13] Islam, M. M., Zerine, I., Rahman, M. A., Islam, M. S., & Ahmed, M. Y. (2024). AI-driven fraud detection in financial transactions. SSRN Working Paper. doi:10.2139/ssrn.5287281.
[14] Al-dahasi, E., et al. (2024). Optimizing fraud detection in financial transactions with ML and imbalance mitigation. Expert Systems, e13682.
[15] Lopez-Rojas, E. A., Elmir, A., & Axelsson, S. (2016). PaySim: A financial mobile money simulator for fraud detection. 28th EMSS, 249–255.
[16] Btoush, M. H., Al-Azzeh, J., Al-Qudah, O., & Al-Shorman, B. (2025). Enhancing credit card fraud detection using traditional and DL models. Frontiers in AI, 8, 1643292.
[17] Ke, G., Meng, Q., Finley, T., et al. (2017). LightGBM: A highly efficient gradient boosting decision tree. NeurIPS, 30, 3146–3154.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY