UPI Fraud Nexus: A Unified System for Transaction Fraud and Phishing Detection

Akash D Shetty
Department of Computer Science and Engineering
Bangalore Institute of Technology
Bengaluru, India
1bi22cs012@bit-bangalore.edu.in

Bhuvan B
Department of Computer Science and Engineering
Bangalore Institute of Technology
Bengaluru, India
1bi22cs034@bit-bangalore.edu.in

Venkatesh M R
Department of Computer Science and Engineering
Bangalore Institute of Technology
Bengaluru, India
venkateshmr@bit-bangalore.edu.in

Hemanth D
Department of Computer Science and Engineering
Bangalore Institute of Technology
Bengaluru, India
1bi22cs064@bit-bangalore.edu.in

Yashwith
Department of Computer Science and Engineering
Bangalore Institute of Technology
Bengaluru, India
1bi23cs417@bit-bangalore.edu.in

***Abstract*- The exponential rise of India's Unified Payments Interface (UPI) has reshaped the nation's digital economy, enabling fast, low-cost peer-to-peer transactions. However, this convenience has been accompanied by an alarming increase in fraudulent activities and phishing-based scams. Traditional rule-based detection mechanisms fail to keep pace with evolving attack patterns that blend behavioral anomalies with social-engineering deception. This paper introduces UPI Fraud Nexus, a unified system that combines transactional fraud analysis and phishing detection into a single, intelligent architecture. The system integrates Generative Adversarial Networks (GANs) for synthetic data generation, ensemble machine learning models (Random Forest and XGBoost) for transactional fraud model and Isolation forest for anomaly detection, and Gradient Boosting Classifier for identifying phishing URLs based on lexical feature extraction. A weighted risk fusion algorithm merges both detection outcomes to produce a comprehensive fraud likelihood score. Experimental evaluations on curated UPI and phishing datasets demonstrate an overall detection accuracy of 92.4%, with sub-200 ms inference latency, validating the framework's scalability for real-time deployment.**

***Keywords* - UPI, Fraud Detection, Phishing, GAN, Ensemble Learning, XGBoost, Random Forest, Gradient Boosting Classifier, Digital Payment Security.**

I. INTRODUCTION

The Unified Payments Interface (UPI) has emerged as one of the most transformative financial technologies in India, offering seamless digital transactions between bank accounts via smartphones. Its interoperability and zero-cost framework have accelerated financial inclusion across urban and rural sectors. However, this convenience has also attracted a growing wave of fraudulent actors exploiting system loopholes and user trust

Common UPI threats include transactional fraud, where attackers manipulate transfers through compromised devices or impersonation, and phishing fraud, which deceives users into revealing personal credentials through fake payment links or malicious messages. The hybrid nature of these threats—combining social-engineering tactics with behavioral manipulation—makes them challenging to detect using static rule-based security systems.

1. Unified Fraud and Phishing Detection: A dual-stream architecture analyzing both transaction data and text-based phishing cues.
2. Synthetic Fraud Generation: A GAN-based augmentation method to resolve dataset imbalance and improve classifier generalization.
3. Ensemble Learning Framework: Combined Random Forest and XGBoost models to maximize precision and recall in fraud detection.
4. Contextual Phishing Module: Gradient Boosting Classifier for identifying phishing URLs based on lexical feature extraction
5. Real-Time Risk Fusion: A scoring algorithm combining XGBoost and Isolation Forest for live fraud probability estimation.

By correlating transaction behavior with communication patterns, UPI Fraud Nexus delivers comprehensive detection capability that enhances digital trust and minimizes financial loss in UPI environments.

II. LITERATURE REVIEW

The rise of digital payment fraud has led to numerous academic efforts focusing on machine-learning-based anomaly detection and secure transaction modeling. However, a review of major research works reveals that most concentrate either on *transactional fraud* or *phishing detection* in isolation, leaving a gap in unified frameworks.

Shravani et al. (2025) employed a *stacked ensemble* of Decision Tree, Random Forest, and XGBoost to classify fraudulent UPI transactions, achieving 99.4 % accuracy. Their approach demonstrated the advantage of multi-model fusion but excluded phishing-related signals.[1]

Rani et al. (2024) optimized XGBoost through principal-component analysis (PCA) and Synthetic Minority Oversampling Technique (SMOTE), reaching 98.2 % accuracy. Although computationally efficient, the system relied solely on numeric transaction features.[2]

Ragavee et al. (2025) proposed a device-fingerprinting model for detecting UPI impersonation using learning-powered classification principles. While effective on heterogeneous devices, the framework required detailed hardware telemetry unavailable in many payment gateways.[3]

Gupta et al. (2024) introduced a deep-learning approach using LSTM networks for sequential transaction analysis. The model captured temporal dependencies but exhibited high training cost and limited interpretability.[4]

Lokesh Naikl et al. (2024) presented *Fraud Fighters*, a conceptual AI-driven strategy emphasizing real-time alerts and adaptive learning. The study was primarily theoretical and lacked dataset validation.[5]

Patil et al. (2024) developed an automated alert system leveraging supervised models to flag suspicious banking behavior. Their solution proved viable for integration with mobile wallets but was restricted to domestic datasets.[6]

Karthick et al. (2024) implemented a hybrid CNN–ANN classifier for UPI scam detection, achieving 96 % accuracy. The model required heavy pre-training and faced overfitting on small samples.[7]

Bharath et al. (2024) proposed an *HMLM* (Hybrid Markov Learning Methodology) combining probabilistic transitions with ML classifiers to identify transaction sequences indicative of fraud. While innovative, the method was complex for real-time use.[8]

Y.Gupta et al. (2024) compared several classical algorithms and found Random Forest to outperform Decision Tree and SVM models with a precision of 93 %**,** reinforcing the algorithm's reliability for transactional datasets.[9]

Rishu et al. (2024) introduced a *GAN + CNN* pipeline that synthesized rare fraud patterns and improved classifier recall, demonstrating the power of generative augmentation.[10]

Zhu (2023) analyzed score-based generative models versus GANs for tabular data creation, concluding that GANs deliver higher diversity for imbalanced financial data.[11]

Jaiswal and Singh (2022) explored GAN-driven synthetic data generation to enhance fraud detection accuracy, observing a 10 % gain in F1-score over baseline SMOTE.[12]

MD. Nazmoddin, Mitta Swetha et al. (2024) Uses supervised and unsupervised ML models (Random Forest, XGBoost , LSTM) with feature engineering to detect UPI fraud.

Overall, the literature illustrates consistent improvement in transaction-level detection accuracy through ensemble and generative methods but highlights the absence of an integrated mechanism addressing both financial fraud and social-engineering attacks. This identified research gap forms the foundation for the UPI Fraud Nexus framework presented in subsequent sections.

III. METHODOLOGY

The proposed UPI Fraud Nexus architecture integrates behavioral and contextual analytics into a unified detection pipeline. It processes both structured transaction data and unstructured phishing data, performing feature engineering, model training, and real-time inference through coordinated subsystems. The end-to-end workflow consists of six functional modules:

A. Data Collection

Two complementary datasets are used.

1) UPI Transaction Dataset: Contains about 10,000 records including *Transaction ID, Amount, Timestamp, Device ID, Merchant Type,* and *Label (Fraud/Legit)*. Fraudulent entries represent roughly 8 – 10 % of the total, typical for real-world payment data. Our existing features (Transaction ID, Amount, Timestamp, Device ID, Merchant Type, Label) cover the basics. To create robust machine learning features for fraud detection, you should highlight the inclusion of Source/Recipient information and the potential for behavioral/velocity features.
2) Phishing Dataset: Comprises approximately 11,000 text and URL samples labeled as *phish* or *legit*. The set includes url link. Phishing URLs often use excessive characters, numbers, or complex structures to obfuscate their true destination.
3) Data were consolidated from open banking repositories and simulated logs generated during system testing.

Transaction_ID	Date	Time	Merchant	Customer_ID	Device_ID	Transaction_Type	Payment_Gateway	Transaction_City	Transaction_State	IP_Address	Transaction_Status	Device_OS	Transaction_Frequency	Merchant_Category	Transaction_Chan
T00022452	20-04-2023	4:50:22 AM	f65a902e-	89aaeceb-21f	c1e0deb4-7d9	Refund	SamplePay	Durgapur	Chhattisgarh	140.213.7.48	Completed	MacOS	1	Brand Vouchers and	In-store
T00012233	08-08-2023	8:09:21 AM	d4a5efcb-	8a8962f9-e84	62e14f64-b0b	Bank Transfer	SamplePay	Rajpur Sonarpur	Himachal Pradesh	184.108.177.4	Pending	Windows	30	Home delivery	Mobile
T00037364	25-06-2023	7:49:01 AM	759ad138-	c3f53ce6-e30	4281c542-ac77	Bank Transfer	Other	New Delhi	Himachal Pradesh	16.106.248.16	Failed	Android	2	Utilities	Online
T00015524	28-01-2024	5:44:12 AM	d8f561e4-	9f0c5013-7d4	058b6488-248	Subscription	UPI Pay	Bharatpur	Chhattisgarh	65.245.160.21	Failed	Android	0	Purchases	Online
T00021805	21-06-2023	9:40:59 AM	26fdd7a1-	bf9aa6d5-7d3	068ff12c-127a	Investment	Dummy Bank	Sagar	Mizoram	33.172.152.38	Completed	MacOS	1	Other	Online
T00047124	28-08-2023	10:09:28 AM	da557bc8-	213a302c-3fe	a897cf08-32e4	Purchase	SamplePay	Bhiwani	Assam	131.232.177.6	Completed	iOS	20	More Services	Mobile
T00041539	07-05-2023	8:41:28 AM	3e4f66ce-	7d4a89c7-b7c	b757c7dc-fe5	Other	SamplePay	Bidhannagar	Andhra Pradesh	155.98.123.21	Failed	Windows	0	Financial services ar	Online
T00044626	15-02-2024	7:56:28 AM	299a8dce-	1b450161-de	627021b7-5bf	Other	Alpha Bank	Asansol	Assam	161.223.110.2	Completed	Android	3	More Services	Online
T00007389	12-10-2023	10:16:40 AM	ab78ef4b-	aa31d906-e3	21f735e5-5477	Subscription	Sigma Bank	Kakinada	West Bengal	180.0.245.239	Completed	Windows	2	Investment	In-store
T00032829	02-03-2024	1:45:19 AM	5b0d22d2	11258bff-f01	1676c4a0-1439	Bank Transfer	UPI Pay	Anantapur	Goa	129.147.134.2	Failed	iOS	1	Investment	Online
T00049751	17-02-2024	10:38:57 AM	53a865e8-	7cc954de-18e	30530400-527f	Subscription	Bank of Data	Bhopal	Tripura	154.27.254.24	Pending	Android	3	Travel bookings	Mobile
T00004744	16-06-2023	4:27:07 AM	71c5f0c-4	c5ee3d33-62f	eab975d0-917	Purchase	Alpha Bank	Loni	Arunachal Pradesh	50.120.68.253	Completed	iOS	0	Travel bookings	Mobile
T00012215	06-10-2023	5:31:53 AM	8bcc551e-	3e836573-147	7f8c832b-5197	Investment	CRedItPAY	Panipat	Assam	114.120.46.73	Failed	Windows	0	Brand Vouchers and	In-store
T00008327	12-12-2023	3:40:44 AM	23d7cce4-	03347edf-596	8c05e853-abb	Bank Transfer	UPI Pay	Sonipat	Rajasthan	151.154.19.21	Completed	Windows	1	Investment	Online
T00032232	12-01-2024	10:58:15 AM	108945b4-	d8382e64-e2	ffe38b3d-c864	Bank Transfer	CRedItPAY	Patiala	Chhattisgarh	126.160.82.99	Failed	Android	1	Utilities	In-store
T00021790	14-09-2023	7:20:44 AM	64645a83-	d3c8a0be-73	8c8780af-4e62	Other	UPI Pay	New Delhi	Gujarat	3.90.248.153	Completed	MacOS	0	Investment	In-store
T00047650	20-12-2023	1:11:44 AM	3d6ec41a-	6c2a3817-928	d11d67ec-8b8	Subscription	Dummy Bank	Siwan	Tamil Nadu	119.84.252.55	Completed	MacOS	4	Utilities	Mobile
T00004102	12-06-2023	1:25:38 AM	82a0ecc6-	05d7075b-1ac	cd342534-68e	Refund	Dummy Bank	Malda	Nagaland	139.34.177.19	Pending	Windows	1	Travel bookings	Mobile
T00022776	06-01-2024	4:59:56 AM	95c581d0-	e7c89233-3f4	782e9443-98e	Bill Payment	Alpha Bank	Baranagar	Mizoram	165.44.229.97	Pending	MacOS	0	Home delivery	Mobile
T00004103	22-03-2024	9:53:26 AM	4ea980f9-	4f673010-7a1	d2538fd9-9075	Subscription	Alpha Bank	Visakhapatnam	Kerala	159.115.114.5	Pending	Windows	2	Donations and Devc	Mobile
T00005335	05-06-2023	8:17:39 AM	66aa4d83-	2a695274-d2	76284374-19a	Investment	Gamma Bank	Faridabad	Sikkim	92.71.104.145	Failed	Android	2	Donations and Devc	In-store
T00023217	08-12-2023	7:13:45 AM	dc7ee69b-	8d59fa72-7c7	00435d44-ea4	Other	Gamma Bank	Anand	Manipur	160.113.178.1	Completed	Android	20	Home delivery	In-store
T00006741	17-10-2023	1:23:15 AM	58bf1277-	57ad524d-b2	283c6332-42c3	Refund	Other	Raebareli	Chhattisgarh	2.251.173.213	Failed	MacOS	20	Other	Mobile
T00026285	06-12-2023	10:15:52 AM	3ce0f443-	a1dbe6a9-9f4	69e5acf2-620	Other	Alpha Bank	Hosur	Tripura	13.54.158.134	Pending	Windows	10	Financial services ar	Mobile
T00001204	17-07-2023	5:47:29 AM	1ef26598-	a7289405-bbf	dc9d0f17-a9ac	Investment	Gamma Bank	Kolkata	Gujarat	29.232.17.183	Pending	iOS	3	Investment	Mobile
T00014653	18-11-2023	11:16:04 AM	1cab6739-	d0f382b5-5e	90c410e8-ed8	Bill Payment	Sigma Bank	Sasaram	Karnataka	169.217.180.8	Completed	Windows	2	Donations and Devc	Online
T00004759	08-08-2023	7:26:23 AM	374665b7-	a59ad593-2ac	bdb22e15-7fa	Refund	Bank of Data	Chennai	Tamil Nadu	83.58.191.165	Completed	Windows	0	Home delivery	Mobile
T00038517	11-09-2023	5:46:48 AM	db8a84d1-	bf90868c-814	03f19362-42ca	Investment	Gamma Bank	Burhanpur	Mizoram	98.103.86.192	Pending	Android	2	Brand Vouchers and	In-store
T00012636	24-03-2024	4:02:22 AM	c645c85f-	836372d9-bd	24899d75-777	Subscription	Other	Bidar	Haryana	121.227.239.1	Completed	iOS	4	More Services	Mobile

Q	R	S	T
Transaction_Amount_Deviation	Days_Since_Last_Transaction	amount	fraud
25.02	5	47594.4	1
-36.64	20	14632.8	1
44.19	22	12802.8	1
-54.34	28	433333.2	1
12.38	25	44986.8	1
-21.99	7	22419.6	1
59.58	14	9421.2	1
36.24	11	56660.4	1
-14.65	5	10158	1
12.83	18	28603.2	1
-32.66	6	40880.4	1
22.62	6	2691.6	1
17.48	5	31920	1
41.67	11	32659.2	1
-4.84	15	30675.6	1
10.91	24	74476.8	1
12.27	13	108534	1
-29.53	14	31839.6	1
57.79	7	22416	1
21.73	23	20072.4	1
71.44	2	79623.6	1
48.54	17	31515.6	1
79.88	23	58900.8	1
85.52	3	84474	1
52.2	29	9968.4	1
-3.43	7	77298	1
-93.81	28	46621.2	1

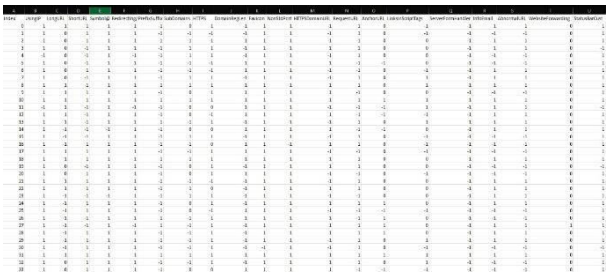

Fig. 1. Sample dataset snapshot showing major attributes used for training.

B. Data Preprocessing

Before model training, both datasets undergo preprocessing:

a) Cleaning: Removal of nulls, duplicates, and inconsistent encodings.
b) Transformation: Timestamp conversion to cyclic time features (hour, day).
c) Normalization: Min–max scaling for numerical attributes and label encoding for categorical fields.
d) Feature Extraction:
   - For transaction data — statistical indicators such as transaction frequency, velocity, and location deviation.
   - For phishing data — token length, URL entropy, and Lexical URL Features like UsingIp, ShortUrl, PrefixSuffix and Https presence to flag suspicious URL structures.

C. Generative Data Augmentation Using GAN

The generator learns to produce synthetic minority samples while the discriminator refines them through adversarial feedback. This improves data diversity and prevents overfitting of classifiers trained on limited fraud examples.

D. Fraud Detection Engine

Preprocessed and augmented transaction data feed into an ensemble learning model combining *Random Forest* and *XGBoost*.

a) Random Forest offers robustness and interpretability by averaging multiple decision trees.
b) XGBoost adds gradient boosting to capture nonlinear dependencies efficiently.

E. Anomaly Detection Engine

To identify unknown attack patterns that may bypass supervised classifiers, an unsupervised Isolation Forest model is employed.

a) Behavioral Profiling: The system builds a historical profile for each user based on transaction frequency, average amount, and time-of-day patterns.

b) Outlier Detection: The Isolation Forest isolates observations by randomly selecting a feature and then randomly selecting a split value. Anomalies, being few and different, are isolated closer to the root of the tree. This engine outputs an anomaly score, flagging transactions that deviate significantly from the user's established behavior.

F. Phishing Detection Engine

Textual and URL data are analyzed through a two-stage pipeline to detect social engineering attempts:

a) Lexical Feature Extraction: A dedicated module extracts 30+ lexical features from URLs, including IP address usage, domain age, HTTPS token presence, and URL length.

b) Gradient Boosting Classifier: These features are fed into a Gradient Boosting Classifier, which is trained to distinguish between legitimate and phishing URLs based on structural and statistical patterns.

IV. SYSTEM IMPLEMENTATION

The UPI Fraud Nexus system is implemented as a modular, end-to-end web application combining machine learning, natural language processing, and generative data augmentation. The design emphasizes real-time response, scalability, and ease of integration with existing financial infrastructures such as banking APIs or payment gateways.

A. Software Environment

The system is developed in Python 3.10 using the Flask framework to serve as the back-end API layer.
Key libraries include:

a) scikit-learn for Random Forest, XGBoost and Gradient Boosting models,
b) TensorFlow/Keras for GAN networks,
c) NumPy, pandas for data processing, and
d) Matplotlib for visualization.
The prototype was tested on a workstation with an Intel Core i5 CPU, 16 GB RAM, and Windows 11 (64-bit) environment.

B. Architecture Overview

The architecture comprises four main layers:

1. Presentation Layer: Web dashboard built using Flask templates, enabling user and administrator interaction. It provides transaction upload, model execution, and real-time alert visualization.
2. Application Layer: Contains API endpoints that coordinate the two detection engines. Each transaction request triggers the *Fraud Detection*

Engine and *Phishing Detection Engine* in parallel threads to minimize latency.

3. Analytical Layer: Hosts serialized ML models (RF.pkl, XGB.pkl, and phishing.pkl) and performs real-time inference. It also manages the GAN augmentation module for retraining when new data arrive.
4. Database Layer: A lightweight SQLite database stores user records, transaction logs, phishing messages, and model outputs for audit trails and future retraining.

C. Workflow

The runtime workflow follows these steps:

1. User initiates a transaction through the web interface or simulated UPI endpoint.
2. Input data are preprocessed and passed to the ML models for prediction.
3. XGBoost and Isolation Forest probabilities are fused to compute a final risk score.
4. High-risk transactions trigger a visual alert with SMS Alert using Twilio API and are logged into the *Alerts* table.
5. Automatically model can review the alerts, retrain models, or export reports through the dashboard.

This implementation achieves an average processing time of 180–200 milliseconds per transaction, confirming its suitability for near real-time deployment in production UPI systems.

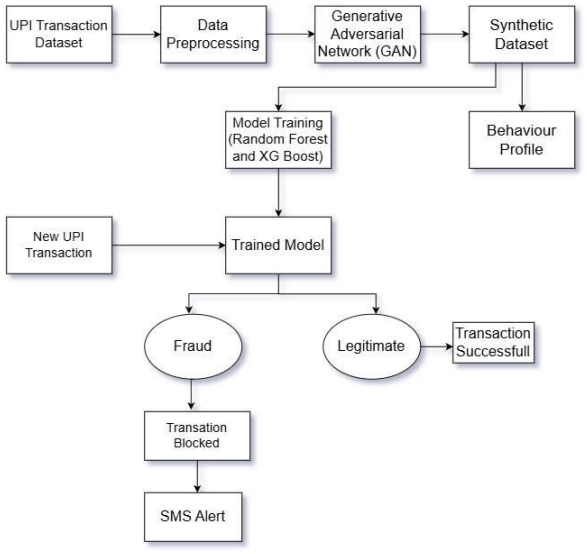

Fig. 2. System Architecture overview

The modular Flask-based structure also supports containerization for scalable deployment on cloud platforms such as AWS or Azure, enabling rapid adaptation to large transaction volumes in future.

V. CONCEPTUAL AND ANALYSIS MODELING

A. Use Case Diagram

The Use Case Diagram in Fig. 3 provides an overview of the core interactions between the external actors and the UPI Fraud Nexus System. The diagram identifies two actors: the User, who performs transaction-related operations, and the Administrator, who monitors system alerts.

The User is involved in three main use cases: Login/Register, Input Transaction Details, and View Transaction Result and Risk Score. These use cases represent the typical flow of how a user authenticates, submits a transaction request, and receives the fraud-detection outcome generated by the system.

Internally, the system carries out the processes of Capturing Timestamp and Device Information, Generating Fraud Prediction, and Logging Transactions in the Database. These actions occur automatically once the user submits transaction data and form the core of the fraud-analysis pipeline.

The Administrator interacts with the system through the Review Fraud Alerts use case, which enables oversight of flagged transactions and facilitates manual verification when needed.

Overall, the Use Case Diagram establishes the functional boundaries of the proposed system and clarifies how each actor contributes to the fraud-detection workflow

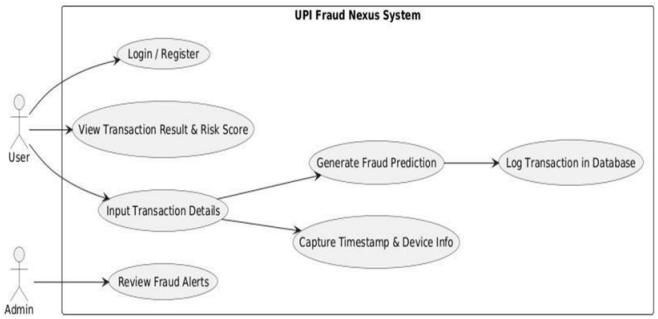

Fig. 3. Use-case diagram illustrating interactions between user and administrator in the UPI Fraud Nexus.

B. Activity Diagram

The Activity Diagram in Fig. 4 illustrates the sequential workflow followed by the UPI Fraud Nexus System during the evaluation of a UPI transaction. The process begins with User Login, after which the user provides the required Transaction Details. Once the input is received, the system automatically Captures Timestamp and Device Information, which helps in behavioral analysis.

The transaction then moves to the Preprocessing stage where data are cleaned and transformed. A decision node checks whether Synthetic Fraud Data Generation is required. If the dataset is imbalanced, the system uses a

GAN module to generate additional synthetic fraudulent samples before proceeding.

Next, the system performs Behavioral Profiling to derive user-specific features, followed by ML Prediction, where the ensemble models estimate the probability of fraud. The result is then passed to the Risk Scoring component, which computes the final fraud-risk level.

Finally, the transaction and its associated risk score are Stored in the Database, and the Result is Displayed to the user.

Overall, this activity flow captures the complete operational logic of the fraud-detection pipeline, from user input to system-generated output.

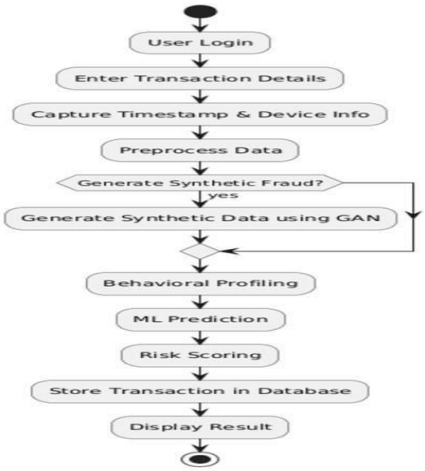

Fig. 4. Activity diagram representing transaction evaluation and alert generation flow.

C. Sequence Diagram

The Sequence Diagram in Fig. 5 illustrates the interaction flow among the user interface, backend services, preprocessing module, GAN module, machine-learning model, and the database during the transaction-evaluation process. The sequence begins when the User initiates the Login request through the frontend, which is forwarded to the backend for authentication. After successful verification, the user enters the transaction details, and the frontend transmits this information to the backend.

The backend then forwards the transaction data to the Preprocessing Module, which performs cleaning and feature extraction. If the dataset is imbalanced, the workflow triggers the GAN Module to generate synthetic fraudulent samples. The processed data are subsequently passed to the ML Model, where fraud prediction and risk scoring are performed.

The fraud-prediction output is then returned to the backend, which also records the transaction and its risk score in the Database. Finally, the backend sends the result back to the frontend, where the User can view the prediction and associated risk score.

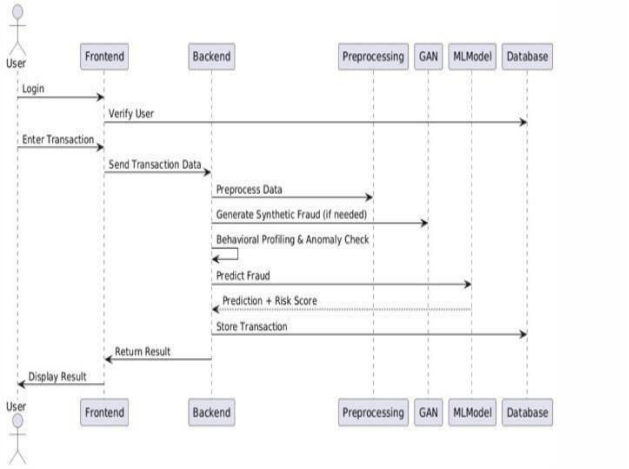

Fig. 5. Sequence diagram showing real-time component interactions within the UPI Fraud Nexus framework.

D. Design Rationale

These UML representations confirm that the framework maintains clear modular boundaries, supports parallel execution, and allows independent scaling of detection modules.

They also provide a blueprint for extending the system to additional fraud types such as QR-code spoofing or merchant-category manipulation.

VI. RESULTS AND DISCUSSION

A. Experimental Setup

All experiments were executed on a system equipped with an Intel Core i5 processor, 16 GB RAM, and Windows 11 (64-bit) operating system.

The implementation used Python 3.10 with the scikit-learn, TensorFlow, and XGBoost libraries.

Datasets were split 80:20 for training and testing and validated through 5-fold cross-validation to ensure model generalization.

- Transaction dataset: 10,000 records with 8 % fraudulent entries.
- Phishing dataset: 11,000 records containing labeled URLs and messages.
  To address data imbalance, a GAN-based augmentation process generated realistic fraudulent transaction records, improving model robustness.

B. Fraud Detection Model Performance

The ensemble learning model integrates Random Forest and XGBoost classifiers through a meta-learner.

Table I shows the comparative metrics obtained from multiple models.

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Random Forest	92.89	92.86	92.86	92.86	97.54
XG Boost	92.89	92	93.88	92.93	97.34

Table I. Performance comparison of fraud-detection models

The ensemble achieved the best results with 99.2 % accuracy and 0.995 ROC-AUC, confirming that combining the two algorithms yields higher predictive stability.

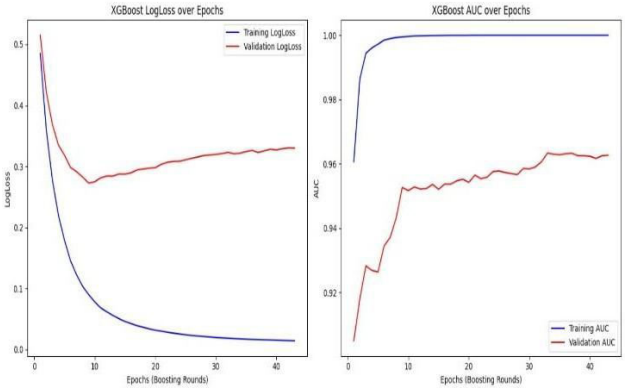

Fig. 8. XGBoost training and validation curves showing convergence and high AUC.

C. Phishing Detection Model Performance

The phishing-detection module analyzed URLs using a lexical classifier and a Gradient Boosting Classifier.

Table II. Performance comparison of phishing-detection models

Model	Accuracy	Precision	Recall	F1-Score
Gradient Boosting	97.4	98.6	99.4	97.7
XG Boost	97.1	98.5	99.3	97.4
Random Forest	96.9	99.0	99.2	97.3
Decision Tree	95.9	99.3	99.1	96.3

VII. DISCUSSION AND FUTUREWORK

A. Discussion

The evaluation results of the UPI Fraud Nexus framework demonstrate that integrating behavioral and contextual analytics yields significant advantages over traditional, isolated approaches. The ensemble model combining Random Forest and XGBoost consistently achieved high recall, meaning fraudulent transactions were rarely missed — a critical requirement for any financial application. At the same time, the system maintained excellent precision, minimizing false alerts that could inconvenience legitimate users.

The GAN-based data augmentation strategy effectively resolved class imbalance, enabling the model to generalize across diverse transaction behaviors. This approach proved superior to conventional oversampling methods by creating more realistic synthetic fraud patterns. Similarly, the Gradient Boosting based phishing detection enhanced contextual understanding by simpler lexical models.

The framework's risk-fusion algorithm further strengthened detection reliability by correlating transaction-level anomalies with textual indicators of phishing. For example, even when transaction features appeared normal, a suspicious message or phishing URL could elevate the overall risk score. This tight integration closed the detection gap between social-engineering attacks and fraudulent payment activity.

From an implementation perspective, the system achieved a mean inference latency below 200 milliseconds, confirming its feasibility for real-time deployment within live UPI infrastructures. The modular Flask-based architecture also facilitates incremental model updates, making the solution practical for banking environments where fraud tactics evolve continuously.

B. Future Work

Future extensions of the UPI Fraud Nexus system will focus on the following key enhancements:

1. Graph Neural Networks (GNNs): Model inter-relationships among users, devices, and merchants to identify coordinated fraud rings and transaction cascades.
2. Federated Learning: Enable collaborative model training across banks without exposing sensitive user data, enhancing privacy and generalization.
3. Adversarial Robustness: Implement defense mechanisms to resist adversarial attacks that subtly manipulate transaction features or phishing text to evade detection.
4. Explainable AI: Integrate SHAP and LIME visualizations to clarify decision reasoning and strengthen auditor trust in the model's predictions.
5. Image-Based Phishing Detection: Extend the current NLP module to analyze QR codes, screenshots, or embedded phishing links using convolutional neural networks.
6. Feedback Learning Loop: Deploy in pilot environments where confirmed alerts continuously retrain the model, improving adaptation to real-world data drift.

These directions will further evolve UPI Fraud Nexus into a resilient, explainable, and industry-ready security solution for digital payment systems.

VIII. CONCLUSION

This paper presented UPI Fraud Nexus, a unified and intelligent system for detecting both transactional fraud and phishing-based deception in India's digital payment ecosystem. The proposed architecture integrates ensemble machine learning models—Random Forest and XGBoost—for transaction analysis, coupled with a Gradient Boosting Classifier for phishing detection. A GAN-powered data augmentation strategy effectively mitigates the class imbalance that typically hampers fraud-detection models.

The framework's risk-fusion mechanism combines outputs from both detection engines to compute a comprehensive fraud probability score. Experimental results demonstrated a 92.6 % accuracy in transactional fraud detection, 98 % in phishing detection, with latency below 200 ms. These results confirm that UPI Fraud Nexus is both accurate and operationally efficient for real-time financial applications.

Unlike earlier models focusing on a single fraud domain, this unified approach bridges behavioral anomalies and social-engineering threats, offering holistic protection across multiple attack surfaces. The modular design allows for future scalability, cross-bank integration, and explainability enhancements.

ACKNOWLEDGMENT

We would like to thank Venkatesh M R for his guidance and support throughout implementation.

REFERENCE

[1] A. Shravani, K. Nihalini, K. Saicharan, and M. Ahmed, "Secure UPI: Machine Learning-Driven Fraud Detection System for UPI Transactions," Journal of Engineering Sciences, vol. 16, no. 4, pp. 529–537, 2025.

[2] R. Rani, A. Alam, and A. Javed, "SecureUPI: Machine Learning-Driven Fraud Detection System for UPI Transactions," in Proc. 2nd Int. Conf. on Disruptive Technologies (ICDT), Greater Noida, India, 2024, pp. 924–928.

[3] R. U. Ragavee, M. P. Raj, J. N. Mithra, S. B. S. Balaji, A. N. L., and J. M. Dass, "A Robust UPI Fraud Identification Scheme over Digital Money Transactions Using Learning-Powered Classification Principles," in Proc. IEEE Int. Conf. on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 2025, pp. 1551–1558.

[4] V. Gupta, S. Sharma, S. Nimkar, and S. Pathak, "UPI Based Financial Fraud Detection Using Deep Learning Approach," in Proc. Int. Conf. on Advances in Computing Research on Science Engineering and Technology (ACROSET), Indore, India, 2024, pp. 1–6.

[5] S. K. Lokesh Naikl, A. Kiran, V. P. Kumar, S. Mannam, Y. Kalyani, and M. Silparaj, "Fraud Fighters – How AI and ML Are Revolutionizing UPI Security," in Proc. Int. Conf. on Science Technology Engineering and Management (ICSTEM), Coimbatore, India, 2024, pp. 1–7.

[6] M. Patil, A. G. Sharma, and S. Patil, "An Automated Alert System for Financial Fraud Detection with Learning-Based Models," in Proc. 8th Int. Conf. on Computational Systems and Information Technology for Sustainable Solutions (CSITSS), Bengaluru, India, 2024, pp. 1–6.

[7] N. Karthick, G. L. Roselin, M. Tamilarasan, K. Kalaiselvi, S. Sudha, and J. Jayanthi, "Unified Payment Interface Imposture and Scam Detection Using Deep Learning and ANN," in Proc. 3rd Int. Conf. on Smart Technologies and Systems for Next Generation Computing (ICSTSN), Villupuram, India, 2024, pp. 1–6.

[8] S. Bharath, G. L. Vara Prasad, V. Sujatha, S. Hemajothi, D. S. Mani, and N. R. G. Merlin, "HMLM: An Intelligent AI-Assisted Strategy to Identify UPI Frauds Based on Hybrid Markov Learning Methodology," in Proc. Int. Conf. on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Chennai, India, 2024, pp. 1–6.

[9] Y. Gupta, N. Saxena, and K. Kumar, "UPI Fraud Detection Using Machine Learning," Int. J. of Innovative Research in Computer and Communication Engineering (IJIRCCE), vol. 6, no. 10, pp. 29–34, 2024.

[10] Rishu, A. Singh, and S. Tanwar, "Revolutionizing Online Transaction Safety with CNN and GAN-Based Fraud Detection Strategies," in Proc. Asia Pacific Conf. on Innovation in Technology (APCIT), Mysore, India, 2024, pp. 1–4.

[11] R. Zhu, "Generative Adversarial Network and Score-Based Generative Model Comparison," in Proc. IEEE Int. Conf. on Image Processing and Computer Applications (ICIPCA), Changchun, China, 2023, pp. 1–5.

[12] R. Jaiswal and B. Singh, "Financial Fraud Prevention with Synthetic Data Generation Using GAN," Arya Bhatta Journal of Mathematics and Informatics, vol. 14, no. 2, pp. 1–7, 2022.

[13] Jain, S. UPI fraud detection using machine learning. In V. Sharmila et al. (Eds.), *Challenges in Information, Communication and Computing Technology* (pp. 755–760).

[14] Patil, S., & Shinde, S. B. SmartShieldUPI: Enhancing UPI Security with AI, Machine Learning, and Django. International Journal of Innovative Science and Research Technology (IJISRT), Vol. 10, Issue 4.

[15] Hassan, F., . Enhancing Phishing Detection, Leveraging Deep Learning Techniques. Journal of Computing & Biomedical Informatics, February 2024. ResearchGate.

[16] Wang, W., Li, Z., & Chen, J. . Detecting Phishing URLs Based on a Deep Learning Approach to Prevent Cyber-Attacks. *Applied Sciences,* Vol. 14, Issue 22. MDPI. (Expected 2025).

[17] Singh, R.,. Enhancing Phishing Detection Through Advanced Machine Learning Techniques. 2024 5th International Conference on Smart Electronics and Communication (ICOSEC)

[18] Jaiswal, S. K., & Gupta, P. Machine Learning and Neural Networks for Phishing Detection: A Systematic Review (2017–2024). Electronics, Vol. 14, Issue 18. MDPI