

Privacy-Aware AI-Based Fraud Detection and Prevention Framework for Secure UPI and Digital Banking Transactions in India

Dr. Shilpa Sarvaiyya¹, Samiksha Sandip Kankale², Ansh Devidas Gupta³

Head of MCA Department, Vidya Bharati Mahavidyalaya, Amravati, Maharashtra, India ¹

Student, MCA Department, Vidya Bharati Mahavidyalaya, Amravati, Maharashtra, India ²

Student, MCA Department, Vidya Bharati Mahavidyalaya, Amravati, Maharashtra, India ³

Abstract: *The Unified Payments Interface (UPI) has made instant, low-friction payments a daily habit for hundreds of millions of people in India, and in doing so has created a large and fast-moving attack surface for fraud. Effective defence needs machine learning over rich behavioural, device and network signals, yet pooling such data conflicts with the privacy obligations placed on banks and payment service providers by India's Digital Personal Data Protection Act, 2023 and by sectoral regulation. This paper proposes PRISM-UPI, a privacy-aware, five-layer framework for real-time fraud detection and prevention. The framework combines (i) privacy-by-design data minimisation and keyed pseudonymisation, (ii) behavioural, device and graph feature engineering, (iii) a tiered detection engine made up of a cost-sensitive stacking ensemble, a graph neural network for mule-account rings and an unsupervised detector for cold-start behaviour, (iv) cross-institution federated learning with secure aggregation and differentially private updates, and (v) a risk-based prevention layer that selects among allow, warn, step-up authentication, delay and block using an expected-cost rule, with SHAP-based explanations and audit trails. We formalise the threat model, specify the learning and privacy mechanisms, and define a reproducible evaluation protocol based on temporal splits, precision-oriented metrics and privacy-utility trade-off analysis. The paper is a design and methodology contribution: because real UPI transaction data is not publicly available, empirical results are not reported here, and the protocol is written so that institutions with data access can reproduce and complete it..*

Keywords: UPI, digital payments, fraud detection, federated learning, differential privacy, explainable AI, class imbalance, graph neural networks, DPDP Act

I. INTRODUCTION

Real-time retail payments in India have shifted decisively towards UPI, which lets a customer move money between bank accounts using a virtual payment address (VPA), a QR code or a phone number, usually in a few seconds and without a card. That convenience is also what attackers exploit. Irrevocability and speed mean that once a fraudulent payment settles, recovery is slow and uncertain, so the realistic window for prevention is the moment of authorisation. Fraud attempts are mostly social-engineering driven rather than purely technical: the victim is persuaded to approve a collect request, scan a malicious QR code, install a remote-access application or share credentials.

Machine learning is the natural tool for screening this volume of traffic in real time. The literature on card and mobile-money fraud shows that three problems recur: extreme class imbalance, shifting fraud behaviour and domain shift between markets, and the need for decisions that investigators and regulators can understand [2]–[5]. A fourth problem is specific to the Indian setting and is the focus of this paper. The most informative signals (payee relationships, device history, location patterns, free-text remarks) are personal data, they are fragmented across hundreds of banks and



payment apps, and their processing is now governed by the Digital Personal Data Protection (DPDP) Act, 2023 [1]. A framework that achieves accuracy by centralising raw personal data is therefore hard to defend legally and carries its own breach risk.

We therefore ask how an AI-based fraud-prevention system for UPI and digital banking can be built so that privacy is a structural property of the design rather than an afterthought, without giving up the precision and latency that real-time payments require. The contributions of this paper are:

A fraud taxonomy and threat model for UPI and digital banking that links each attack pattern to the signals and prevention actions that address it.

PRISM-UPI, a five-layer architecture that integrates data minimisation, a tiered hybrid detector, federated learning with secure aggregation and differential privacy, and a risk-based prevention policy.

An expected-cost decision rule that turns calibrated risk scores into graduated interventions, so that friction is applied in proportion to risk and false positives are priced explicitly.

A reproducible evaluation protocol covering temporal validation, precision-centred metrics, latency, privacy–utility trade-offs and fairness checks.

The remainder of the paper is organised as follows. Section II reviews related work. Section III defines the threat model and problem. Section IV presents the framework. Section V gives the evaluation protocol. Section VI discusses deployment and regulatory alignment, and Section VII concludes with limitations and future work.

II. RELATED WORK

A. Class Imbalance and Ensemble Learning

Fraud labels are rare, so classifiers trained naively favour the majority class. Resampling methods such as SMOTE [7] and its cluster-based and cleaning variants are common remedies. Damanik and Liu [2] combine K-means clustering, SMOTE and Edited Nearest Neighbours with a stacking ensemble on the synthetic PaySim data [6] and report very high scores, while themselves cautioning that cleaned synthetic data can lead to overfitting and that training cost on the full data set is high. Two lessons carry into our design: ensembles that combine diverse learners (stacking [10], gradient boosting [8], [9]) are strong baselines for tabular transaction data, and resampling should be applied only inside training folds and validated on temporally later, untouched data.

B. Transfer Across Markets and Domain Shift

Lebichot et al. [3] study how a fraud model built for one country can be adapted to another, comparing fifteen transfer strategies on more than two hundred million e-commerce transactions. They find that the best strategy depends on how many target-domain labels are available: generative self-supervised adaptation is competitive when labels are scarce, and supervised or augmented-feature methods win once labels accumulate, with a weighted ensemble of both performing best overall. This is directly relevant to India, where a new payment app, a new bank or a new customer segment resembles a new target domain, and where federated aggregation can be seen as a privacy-preserving way of sharing source-domain knowledge.

C. Multimodal Evidence and Explainability

Ping et al. [4] fuse numerical, textual and layout evidence for financial-statement fraud and attach a human-readable evidence chain to each prediction. The setting differs (annual filings rather than payments), but the principle transfers: signals from different sources should be combined, and the system should say which signals drove a decision. In payments the analogous modalities are transaction features, behavioural and device context, graph structure and the free-text remark field. For explanation we rely on SHAP [21], which provides additive per-feature attributions suitable for audit logs.

D. Privacy-Preserving Learning

Federated learning trains a shared model by exchanging model updates instead of raw data [15], [16]. Updates can still leak information about individual records, so federated learning is normally combined with secure aggregation [19], which hides individual updates from the server, and with differential privacy [17], applied to training through clipped,



noised gradients [18] and tracked with Rényi accounting [20]. These tools are mature in the research literature but are rarely specified end-to-end for payment-fraud pipelines with real-time constraints, which is the gap this paper addresses.

E. Graph and Anomaly Methods

Money-mule networks are relational: many victims pay a few accounts that quickly forward funds. Graph neural networks such as GCN [12] and GraphSAGE [13] learn from this structure, and GraphSAGE's inductive formulation suits graphs that change continuously. Unsupervised detectors such as Isolation Forest [11] cover cases with no labelled history, for example a newly onboarded account.

III. THREAT MODEL AND PROBLEM FORMULATION

A. UPI Fraud Taxonomy

Table I summarises the principal fraud patterns the framework targets. The listed signals are examples of what a detector can observe at authorisation time; they are design assumptions and not statistics.

TABLE I: Fraud Taxonomy, Discriminating Signals and Preventive Action

Pattern	Mechanism	Example signals	Preventive action
Collect-request / refund scam	Victim is tricked into approving a pull request while believing they will receive money	Unsolicited collect from new payee, urgent remark text, payer is not the usual approver	Explicit warning on screen; step-up authentication
Malicious QR / payment link	QR or link initiates a debit, not a credit	QR from unverified merchant, mismatch between claimed and actual payee	Payee-verification prompt; hold on first-time large payment
Vishing and remote-access takeover	Caller persuades victim to install screen-sharing app or reveal credentials	Remote-session indicators, new device binding, rapid sequence of changes in limits	Block risky device actions; cooling-off period
SIM swap / device cloning	Attacker re-binds the account to a new SIM or device	New device plus SIM change, behavioural mismatch, unusual login time	Step-up authentication; delayed activation of high limits
Mule accounts and layering	Funds are fanned in and quickly forwarded out	High in-degree, short dwell time, many first-time senders, shared device	Graph-based hold; account review
Account takeover in net banking	Stolen credentials used for new beneficiary and transfer	New beneficiary followed by large transfer, atypical session behaviour	Beneficiary cooling-off; step-up authentication

B. Adversary and Privacy Requirements

We consider an external adversary who conducts social-engineering and account-takeover attacks and who adapts to the detector over time (concept drift, adversarial probing of thresholds). We also consider an honest-but-curious central aggregator and honest-but-curious peer institutions, which should not be able to infer a customer's transactions or counterparties from shared artefacts. Insider misuse and compromise of an institution's own raw data store are out of scope for the learning protocol and are addressed by conventional access control and encryption. The design targets the following privacy requirements: purpose limitation and data minimisation at ingestion, no exchange of raw records between institutions, bounded leakage from shared model updates, and an auditable explanation for every adverse decision.



C. Formal Problem

Let institution k hold a local data set $D_k = \{(x_i, g_i, y_i)\}$, where x_i is a vector of transaction and behavioural features, g_i is the local neighbourhood of the transaction in the payment graph, and y_i is a fraud label that may arrive with delay (for example after a customer complaint or investigation). The goal is to learn a scoring function f that outputs a calibrated risk probability $p = f(x, g)$ within a latency budget, such that (a) the data never leaves institution k in raw form, (b) the shared learning procedure satisfies (ϵ, δ) -differential privacy at the level of the update, and (c) the downstream policy minimises expected cost over fraud losses, customer friction and operational review effort.

IV. THE PRISM-UPI FRAMEWORK

PRISM-UPI (Privacy-preserving, Real-time, Intelligent, Secure Monitoring for UPI) has five layers. Table II gives an overview, and the subsections describe each layer.

TABLE II: Layers of PRISM-UPI

#	Layer	Main function	Privacy mechanism
1	Ingestion and minimisation	Collect only fields needed for fraud purpose; pseudonymise identifiers; generalise sensitive attributes	Keyed hashing, field-level minimisation, retention limits
2	Feature engineering	Behavioural, device, velocity, graph and text features computed in a streaming store	Features computed in-house; no raw identifiers leave the institution
3	Hybrid detection engine	Stacking ensemble, GNN for mule rings, unsupervised cold-start detector, calibrated fusion	Models trained locally or federally
4	Federated learning	Cross-institution training using secure aggregation and DP updates	Secure aggregation, clipping, Gaussian noise, privacy accounting
5	Prevention and governance	Expected-cost decision, graduated friction, SHAP explanations, audit and grievance workflow	Explanation without exposing other users' data; access-controlled logs

A. Layer 1: Privacy-by-Design Ingestion

Only fields with a defined fraud-prevention purpose are admitted to the feature pipeline: amount, timestamp, channel, payee and payer identifiers, device and session indicators, coarse location and the remark field. Direct identifiers such as VPA, phone number and account number are replaced by keyed pseudonyms before they reach analytical stores:

$$t = \text{Truncate}(\text{HMAC}_K(id)) \quad (1)$$

where K is a secret key held in a hardware security module and rotated on a schedule. Pseudonyms are stable within a key epoch, so graph and velocity features remain computable, but they cannot be reversed without the key. Continuous attributes with re-identification risk, such as exact coordinates, are generalised (for example to a coarse area), and free-text remarks are reduced to derived indicators (length, urgency keywords, presence of links) rather than stored verbatim in the training set. Retention periods are tied to the fraud-prevention purpose and to applicable regulation.

B. Layer 2: Feature Engineering

Four feature families are used. Transaction features capture amount, hour of day, payee novelty and amount relative to the customer's own history. Behavioural and velocity features count transactions, distinct payees and cumulative value over sliding windows (for example 1 minute, 1 hour, 24 hours). Device and session features capture device age, binding changes, SIM-change recency and indicators of remote-access tools. Graph features capture in-degree, out-degree, fan-in/fan-out ratio, dwell time of received funds and the number of first-time senders. Following the observation in [2] that merging weakly informative fields can produce more compact and useful representations, categorical combinations



such as payer type by payee type are encoded as single features. As in [4], features are standardised within a relevant stratum, here customer segment and time period, to reduce spurious differences between populations.

C. Layer 3: Hybrid Detection Engine

The detector has three complementary components whose outputs are calibrated and fused.

Tier 1: stacking ensemble. Gradient-boosted trees [8], [9] and a random forest are trained as base learners. Following Wolpert [10], a logistic-regression meta-learner is trained on out-of-fold base predictions so that the meta-learner never sees predictions made on data the base learners were fitted to:

$$p_{ens} = \sigma(w_0 + \sum_m w_m \cdot \text{logit}(p_m)) \quad (2)$$

Imbalance is handled in two ways. Class weighting or focal loss [14] is applied during training:

$$L_{focal} = -\alpha (1-p)^y y \log p - (1-\alpha) p^y (1-y) \log(1-p) \quad (3)$$

and any oversampling such as SMOTE [7] is confined to training folds. Because over- and under-sampling distort the score distribution, Platt or isotonic calibration is fitted on an untouched, time-ordered calibration set before scores are used for cost-based decisions.

Tier 2: graph model. A GraphSAGE-style network [13] aggregates neighbour information in the pseudonymised payment graph to score accounts and edges for mule behaviour:

$$h_v^{(l+1)} = \text{ReLU}(W^{(l)} \cdot \text{MEAN}(\{h_v^{(l)}\} \cup \{h_u^{(l)} : u \in N(v)\})) \quad (4)$$

The inductive form allows new accounts to be scored without retraining. The graph model runs in near-real time on a rolling subgraph, and its output is cached so that the authorisation path reads a precomputed account score.

Tier 3: unsupervised cold-start detector. Isolation Forest [11] or a similar density-based detector scores deviation from an account's own or its segment's behaviour. It is important for new customers, new fraud types and periods when labels lag behind attacks. Its score s_{anom} is rescaled to [0, 1].

The final risk score is a calibrated weighted combination:

$$R = \lambda_1 p_{ens} + \lambda_2 p_{gmn} + \lambda_3 s_{anom}, \quad \sum \lambda_j = 1 \quad (5)$$

with weights fitted on the validation period. The transfer-learning study in [3] found that blending a label-free generative component with a supervised component keeps performance stable when labels are scarce; Eq. (5) applies the same idea, using the unsupervised and graph components as a hedge against label scarcity and delay.

D. Layer 4: Federated Learning with Secure Aggregation and Differential Privacy

Banks and payment service providers each train the Tier 1 and Tier 2 models on local data and share only model updates. In round t , institution k computes an update Δ_k , clips it to norm C , and participates in secure aggregation [19] so that the aggregator sees only the sum. Gaussian noise calibrated to the clipping norm is added, following the DP-SGD recipe [18]:

$$\theta^{(t+1)} = \theta^{(t)} + (1/K) (\sum_k \text{clip}(\Delta_k, C) + N(0, \sigma^2 C^2 I)) \quad (6)$$

A privacy accountant [20] tracks the cumulative (ϵ, δ) budget over rounds. Since fraud data is highly non-IID across institutions (a wallet app, a large retail bank and a small co-operative bank see very different traffic), the framework supports local fine-tuning of a shared global model, which is the federated analogue of the target-domain adaptation that [3] found valuable. Tree ensembles do not train by gradient descent, so for the boosted-tree component the framework uses either a federated variant that shares noised histogram statistics or distils the federated neural model into the local tree model; the choice is left to the implementer and is part of the evaluation protocol. Clients whose update statistics are anomalous are down-weighted or excluded to limit model-poisoning attacks.

E. Layer 5: Risk-Based Prevention, Explanation and Governance

A single block/allow threshold is a poor fit for payments, because the cost of friction differs from customer to customer and fraud losses scale with amount. Let $A = \{\text{allow, warn, step-up, delay, block}\}$ be the set of actions. For each action a we define an expected cost



$$E[\text{cost}(a)] = R \cdot L_{\text{miss}}(a, \text{amt}) + (1 - R) \cdot C_{\text{fric}}(a) \quad (7)$$

where $L_{\text{miss}}(a, \text{amt})$ is the expected loss if the transaction is fraudulent and action a is taken (it is large for allow and small for delay or block), $C_{\text{fric}}(a)$ is the cost of unnecessary friction or a wrongly declined payment, and amt is the transaction amount. The system selects the action with the lowest expected cost. In practice this yields graduated behaviour: low-risk payments pass silently, mid-risk payments get a contextual warning (for instance that a collect request will debit, not credit, the account) or step-up authentication, higher-risk payments to new payees are delayed with a cooling-off period, and only the highest-risk cases are blocked. The cost parameters are policy choices that should be set with risk, compliance and customer-experience teams and reviewed periodically.

Each adverse or intervening decision is accompanied by a SHAP explanation [21] limited to the customer's own features and to graph-level indicators that do not reveal other customers' identities. Investigators see a ranked list of contributing factors in the spirit of the evidence chain of [4], while the customer sees a plain-language reason. Decisions, scores, model versions and explanations are written to an access-controlled audit log. A grievance channel allows customers to contest a decision, and outcomes feed back as delayed labels. Because automated decisions can affect access to payments, fairness is monitored by comparing false-positive rates across customer segments, regions and device types.

F. End-to-End Flow and Latency

At authorisation time the pipeline (i) pseudonymises the request, (ii) retrieves streaming features and the cached graph score, (iii) evaluates the lightweight ensemble and cold-start detector, (iv) applies calibration and Eq. (5), and (v) applies Eq. (7) to choose an action. Heavy computation, namely graph updates, federated training rounds and recalibration, runs off the critical path. Because UPI authorisation operates under tight time limits, the design target is a scoring budget of a few tens of milliseconds at the institution, and the evaluation protocol measures tail latency rather than assuming it.

V. EVALUATION PROTOCOL

No real UPI data is used in this paper. The protocol below specifies how PRISM-UPI should be evaluated by institutions with data access, and what can be done with public data in the meantime.

A. Data

Proprietary: de-identified UPI and net-banking transaction logs with fraud labels from complaints and investigations, ideally from at least two institutions to test federated learning.

Public development data: the PaySim synthetic mobile-money data set [6] for pipeline development. Its synthetic origin and its dependence on a few strong features mean that results on it should not be read as evidence of real-world performance; this caution applies equally to near-perfect scores reported on it in the literature.

B. Splits and Leakage Control

Data is split chronologically into training, validation and test periods, with an additional embargo gap between periods to account for label delay. Resampling, scaling, calibration and feature selection are fitted only on the training or calibration periods. Random shuffling across time is not used, because it inflates performance on drifting fraud data. For federated experiments, institutions are held out in addition to time periods to measure generalisation to an unseen participant, in the manner of the transfer scenarios in [3].

C. Baselines and Ablations

Baselines are logistic regression, a single gradient-boosted model, a random forest, the stacking ensemble alone, a centralised (non-private) model trained on pooled data as an upper bound, and purely local models as a lower bound for federated learning. Ablations remove, one at a time, the graph model, the unsupervised detector, calibration, differential-privacy noise, secure aggregation (to measure its overhead) and the expected-cost policy (replaced by a single threshold).



D. Metrics

Table III lists the metrics and what each is meant to show.

TABLE III: Evaluation Metrics and What Each Is Meant to Show

Metric	Purpose	Note
AUPRC / average precision	Ranking quality under extreme imbalance [22]	Preferred over ROC-AUC, which can look high when positives are rare
Precision@K and recall at fixed false-positive rate	Matches the review capacity and customer-friction limits of an institution	Choose K and FPR from operational constraints
Expected monetary loss and friction cost	Value-weighted outcome of the policy in Eq. (7)	Compare against single-threshold policy
Calibration error (ECE, Brier score)	Validity of probabilities used in cost-based decisions	Report before and after calibration
p50 / p99 latency	Feasibility within payment time limits	Measure on production-like hardware
Privacy budget (ϵ , δ) and membership-inference success	Quantifies leakage from shared updates	Report utility as a function of ϵ
Fairness	Comparison of false-positive rates across customer groups	Investigate any large gaps

All stochastic experiments should be repeated over multiple seeds with confidence intervals, and statistical comparison of methods should use a paired test over matched evaluation windows. Reporting accuracy is discouraged, since a model that approves everything is highly accurate when fraud is a tiny fraction of traffic.

E. HYPOTHESES TO BE TESTED

H1: the tiered engine improves AUPRC and Precision@K over the best single model, with the largest gains on mule-account and cold-start fraud.

H2: federated training with secure aggregation recovers most of the gap between local-only and centralised models, and the loss from differential privacy remains modest at moderate ϵ .

H3: the expected-cost policy reduces combined loss and friction cost compared with a single threshold at equal review capacity.

H4: fine-tuning the federated model locally helps institutions with atypical traffic more than it helps large institutions.

These are hypotheses motivated by the literature [2], [3], [18], not findings of this paper, and should be rejected if the data does not support them.

VI. DEPLOYMENT AND REGULATORY CONSIDERATIONS

The framework is intended to support, not to replace, an institution's legal and compliance analysis. Table IV maps design elements to the regulatory themes that normally apply; the exact obligations should be confirmed against current text and with counsel, since rules and circulars change.

TABLE IV: Mapping of Design Elements to Regulatory Themes (to Be Verified Against Current Rules)

Theme	Relevant source	Design element
Purpose limitation, minimisation, security safeguards, breach handling	DPDP Act, 2023 [1] and its rules	Layer 1 minimisation, keyed pseudonymisation, access-controlled logs, retention limits
Lawful basis and notice for processing for fraud prevention	DPDP Act, 2023 [1]	Documented purpose, notice text, and review of whether a legitimate-use basis applies



Theme	Relevant source	Design element
Storage of payment data in India	RBI directions on payment-system data storage	Training and feature stores kept in India; federated design avoids cross-border movement of raw data
Customer authentication and fraud-risk management in digital payments	RBI and NPCI requirements for digital payments	Step-up authentication and cooling-off actions in Layer 5, aligned to prescribed controls
Incident reporting and redressal	RBI, NPCI and CERT-In frameworks	Audit trail, grievance workflow, incident playbook

Operationally, deployment should proceed in stages: shadow mode (scores logged but not acted on), then low-friction actions such as warnings, then graduated step-up and delay actions, with human review capacity sized from the Precision@K results. Models need drift monitoring, scheduled recalibration and a rollback path. Adversarial probing of thresholds should be countered by randomising some interventions and by not exposing scores to customers or counterparties. Finally, the human side of UPI fraud matters as much as the model: contextual warnings and customer education complement automated screening, because the dominant attacks work by persuading the victim to authorise the payment themselves.

VII. LIMITATIONS AND FUTURE WORK

This paper presents a design and an evaluation protocol, not an empirical result, and its main limitation is the absence of validation on real UPI data, which cannot be released publicly. Several design choices carry known tensions: differential privacy lowers utility for rare events, because clipping and noise affect the small number of fraud examples most; secure aggregation and graph computation add engineering and latency cost; delayed and noisy labels bias supervised training towards fraud that was detected and reported; and cost parameters in Eq. (7) are subjective. Federated learning also does not by itself guarantee legal compliance.

Future work includes running the protocol with one or more institutions, extending the federated component to graph models across institutions without sharing edges, adding a text model for remark fields and voice-phishing transcripts in the spirit of multimodal fusion [4], studying robustness to adaptive adversaries, supporting regional-language remarks, and evaluating customer-facing warning designs through user studies.

VIII. CONCLUSION

Fraud prevention for UPI and digital banking in India has to meet two demands at once: strong, real-time detection and strict respect for the privacy of the people whose transactions are being scored. PRISM-UPI addresses both through minimisation and pseudonymisation at ingestion, a tiered detector that combines supervised, graph and unsupervised evidence, privacy-preserving federated training across institutions, and an expected-cost policy that applies friction in proportion to risk with explanations and audit trails. The accompanying evaluation protocol is designed so that claims about accuracy, latency and privacy can be tested honestly on real data. We hope it serves as a starting point for institutions and researchers building privacy-aware fraud defences for India's digital payments ecosystem.

REFERENCES

1. Government of India, "The Digital Personal Data Protection Act, 2023," Act No. 22 of 2023.
2. N. Damanik and C.-M. Liu, "Advanced fraud detection: Leveraging K-SMOTEENN and stacking ensemble to tackle data imbalance and extract insights," *IEEE Access*, vol. 13, pp. 10356–10370, 2025.
3. B. Lebiclot, T. Verhelst, Y.-A. Le Borgne, L. He-Guelton, F. Oblé, and G. Bontempi, "Transfer learning strategies for credit card fraud detection," *IEEE Access*, vol. 9, pp. 114754–114766, 2021.



4. W. Ping, Y. Jiao, H. Fan, and X. Zhang, "Multimodal fraud detection in financial statements: A trimodal attention network with contrastive evidence chain construction," *IEEE Access*, vol. 14, pp. 80456–80468, 2026.
5. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: A realistic modeling and a novel learning strategy," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3784–3797, 2018.
6. E. A. Lopez-Rojas, A. Elmir, and S. Axelsson, "PaySim: A financial mobile money simulator for fraud detection," in *Proc. 28th Eur. Model. Simul. Symp. (EMSS)*, 2016, pp. 249–255.
7. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.
8. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794.
9. G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.
10. D. H. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, 1992.
11. F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, 2008, pp. 413–422.
12. T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.
13. W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.
14. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2980–2988.
15. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2017, pp. 1273–1282.
16. P. Kairouz et al., "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, no. 1–2, pp. 1–210, 2021.
17. C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory of Cryptography Conf. (TCC)*, 2006, pp. 265–284.
18. M. Abadi et al., "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2016, pp. 308–318.
19. K. Bonawitz et al., "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2017, pp. 1175–1191.
- I. Mironov, "Rényi differential privacy," in *Proc. IEEE Comput. Secur. Found. Symp. (CSF)*, 2017, pp. 263–275.
20. S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 4765–4774.
21. T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS ONE*, vol. 10, no. 3, e0118432, 2015.

