बैंक ऑफ़ बड़ौदा
Bank of Baroda

वित्तीय सेवाएं विभाग
DEPARTMENT OF
FINANCIAL SERVICES

1. Title & Team details
2. Problem statement
3. Outline of Unique & innovative solution
4. Business model and commercialization potential.
5. Proposed process flow / Architecture of implementation
6. Algorithm Details
7. One-page summary of the project.

MuleGuard AI

Ashutosh Anand

aashutosh22@iitk.ac.in

+91 9771060504

Jyotish Singha

jyotishsin22@iitk.ac.in

+91 8135080753

2. Problem statement

Design a graph-based machine learning system to detect mule accounts and mule rings across multiple banks in near real time while preserving privacy and regulatory compliance

Context and Motivation

India's financial ecosystem processes billions of UPI/NEFT/RTGS/IMPS transactions every month. Alongside this growth, mule accounts (bank accounts) used as pass-through conduits for laundering stolen or illicit funds which pose a systemic risk.

Many Rule based algorithms demonstrated the value of AI in detecting mule accounts, achieving high accuracy. However, fraudsters rapidly adapt, spreading transactions across multiple accounts, devices, and banks which makes detection harder.

Recent ORF policy work on "India's AI Imperative" stresses that AI must be applied to protect critical infrastructure and enable responsible, explainable, and privacy-preserving systems. Similarly, the AI-Driven Credit Assessment report highlights challenges banks face in deploying AI like data fragmentation, lack of standardization, privacy concerns, and need for explainability, making this a ripe research and development opportunity.

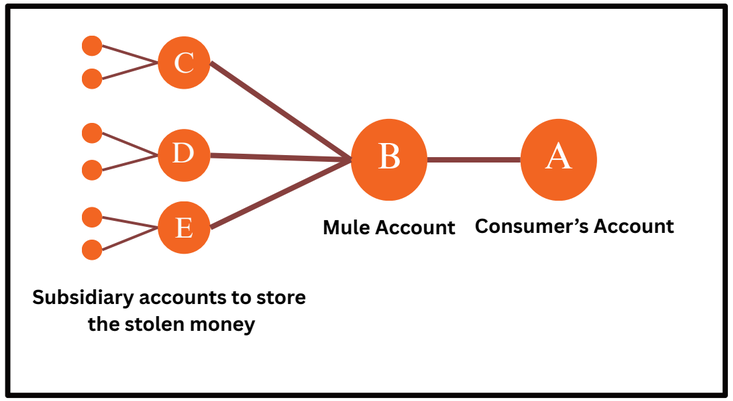
बैंक ऑफ़ बड़ौदा
Bank of Baroda

वित्तीय सेवाएं विभाग
DEPARTMENT OF
FINANCIAL SERVICES

3. Outline of Unique & innovative solution

Mule accounts represent a critical threat to the banking ecosystem as they facilitate the movement of illicit funds across multiple financial institutions, often operating as part of sophisticated rings that exploit the isolation between banks' fraud detection systems.

Our solution can identify both individual mule accounts and organized mule rings while operating across multiple banks simultaneously.



Key inputs

We ingest streaming events from multiple domains:

- core banking transactions (credits, debits, transfers)
- onboarding/KYC information,
- payment logs
- card-present POS data
- digital-channel telemetry (device fingerprints, IP/geolocation, log-in/session events)
- Then we enrich these with call-centre metadata and behavioural biometrics (typing cadence, swipe patterns)

These events are normalised and converted into rolling features such as velocity changes, dormant-to-active shifts, device/IP entropy, merchant concentration and pass-through ratios.

Responsible AI and Privacy

The model uses federated learning to enable collaboration across banks without sharing raw PII. Techniques like secure aggregation, differential privacy, and pseudonymization ensure compliance with regulations (e.g., DPDP Act 2023).

Core concept

Our solution models the entire ecosystem as a heterogeneous graph:

- Nodes represent entities — accounts, customers, devices, IP addresses, merchants, phone numbers, and sessions.
- Edges represent relationships — transactions, logins, device/IP usage, call-center interactions, and temporal links.
- Edge weights capture strength, frequency, recency, and risk of the connection.

A Graph Neural Network (GNN) is trained on this graph to learn embeddings that encode each node's local structure and neighborhood patterns.

Competitive Edge

- Compared to many rule based or siloed systems :
- Captures multi-entity, multi-channel patterns across institutions.
- Learns continuously (concept drift detection + retraining).
- Produces explainable alerts with graph visualizations and feature contributions.
- Maintains strong privacy guarantees while benefiting from network-wide intelligence.

Key Idea

Fraud signals often emerge from relationships rather than isolated attributes. By aggregating information from neighbors (message-passing), the GNN learns patterns like:

- High fan-in/fan-out ratios.
- Device reuse across many accounts.
- Dormant-to-active shocks.
- Short transaction paths forming cycles or rings.

This lets the system flag entire mule rings, not just single suspicious transactions.

4. Business model and commercialization potential.

What we sell ?

- A fraud-detection platform that spots mule accounts and rings in near real time across banks.
- Core pieces: risk scoring API, graph alerts and visuals, analyst dashboard, and privacy-first federation so data never leaves each bank.

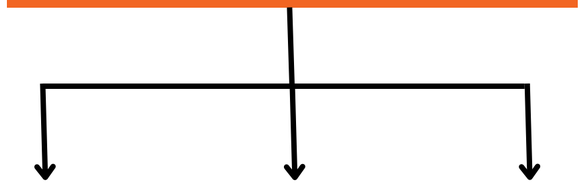
Who buys it?

- Large banks, mid-size banks, payment service providers, and merchant acquirers who see UPI/NEFT volume and rising mule activity.
- Ideal first movers: 2–3 banks willing to co-pilot and share non-PII signals through our federated setup.

How we make money ?

- Subscription: annual platform fee by institution, tiered by transaction volume.
- Per-use: small fee per million transactions scored.
- Add-ons: compliance pack (DPDP logs, audit reports), premium support, and custom rule packs.
- Outcome option: shared-savings model for pilots (a % of fraud loss avoided).

Go-to-market



Phase 1 (0–6 months): two-bank pilot on selected corridors; measure recall, false positives, and money blocked

Phase 2 (6–12 months): expand to 5–7 institutions; add payment aggregators; publish joint results.

Phase 3 (12+ months): "Federated Fraud Network" membership with shared typologies and faster cross-bank response.

Why they pay ?

- Fewer losses: catch organized rings, not just single bad transfers.
- Lower noise: better precision means fewer false alarms and faster case closures.
- Easy to adopt: plugs into existing TMS/AML tools; no raw PII leaves the bank.
- Audit ready: clear reasons for each alert and reports that satisfy risk and compliance teams.

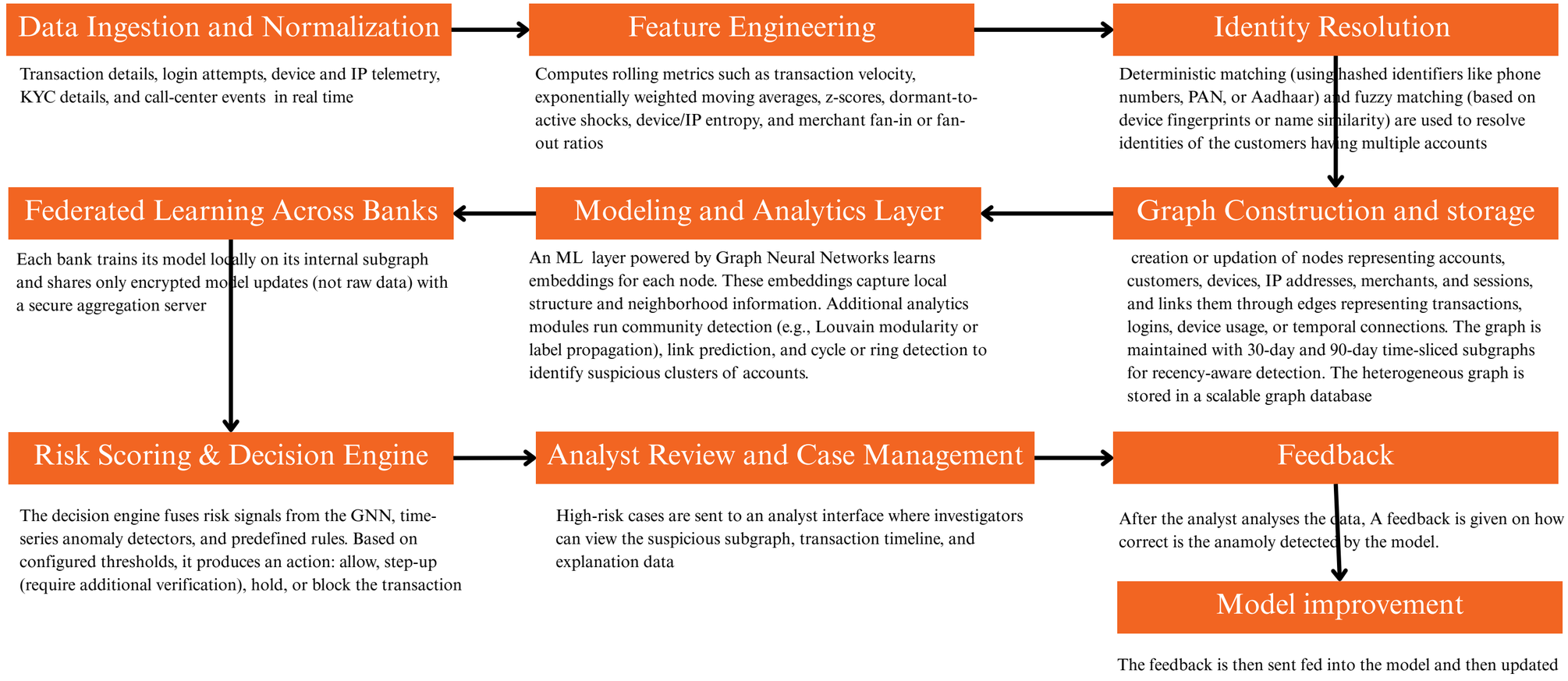
What makes this defensible ?

- Network effect: more members → better ring detection for everyone.
- Privacy moat: federated learning and secure aggregation built-in, easing DPDP reviews.
- Graph know-how: device/IP/merchant link features and ring-finding playbooks that are hard to copy.

बैंक ऑफ़ बड़ौदा
Bank of Baroda

वित्तीय सेवाएं विभाग
DEPARTMENT OF
FINANCIAL SERVICES

5. Proposed process flow / Architecture of implementation



6. Algorithm Details

(in Pseudocode)

ALGORITHM RiskScoringAndDecision (Inputs ->
event_data, # *transaction/session details (amount, account, merchant, device, IP, timestamp)*
entity_map, # *resolved IDs: account_id, customer_id, device_id, ip_id, merchant_id, session_id*
node_features, # *precomputed features for accounts, customers, devices, merchants*
edge_features, # *transaction frequency, recency, device-IP entropy, etc.*
subgraph_context, # *30–90 day neighborhood of focal account with nodes & weighted edges*
model_parameters, # *GNN weights, fusion weights, calibration curves*
policy_thresholds, # *decision thresholds: allow, step-up, hold, block*
domain_rules # *business rules: risk uplifts, STR/SAR triggers*
)

1) Graph Construction
nodes ← CreateOrUpdateNodes(entity_map, node_features)
edges ← CreateOrUpdateEdges(entity_map, edge_features)
ApplyTemporalDecay(edges)
PersistGraph(nodes, edges)

2) Subgraph Preparation
focal ← DetermineFocalTarget(event_data)
subgraph ← FetchSubgraph(focal, subgraph_context)
features ← CollectFeatures(focal, subgraph)

3) Model Scoring
score_gnn ← GNNScore(subgraph, features, model_parameters)
score_ts ← TimeSeriesAnomalyScore(features)
score_rule ← RuleScore(event_data, domain_rules, features)

4) Score Fusion
fused_score ← WeightedFusion({
gnn: score_gnn,
ts: score_ts,
rule: score_rule
}, model_parameters.fusion_weights)

5) Decision Engine

IF fused_score < policy_thresholds.allow THEN
action ← "ALLOW"
ELSEIF fused_score < policy_thresholds.step_up THEN
action ← "STEP-UP AUTHENTICATION"
ELSEIF fused_score < policy_thresholds.hold THEN
action ← "HOLD FOR ANALYST REVIEW"
ELSE
action ← "BLOCK TRANSACTION"
ENDIF

6) Explainability

explanation ← BuildExplanation(
focal,
TopContributors({score_gnn, score_ts, score_rule}),
ImportantGraphPaths(subgraph)
)

7) Case Creation (if escalated)

IF action IN {"STEP-UP AUTHENTICATION","HOLD FOR ANALYST REVIEW","BLOCK TRANSACTION"} THEN
case_id ← CreateCase(focal, fused_score, action, explanation, subgraph)
ENDIF

8) Return Output

RETURN {
action: action,
score: fused_score,
case: case_id or null,
explanation: explanation
}

END ALGORITHM

For more details visit our Github Link: Github Link

7. One-page summary of the project.

MuleGuard AI demonstrates that graph-based machine learning combined with federated learning can effectively identify mule accounts in a privacy-preserving, explainable, and scalable manner. By modeling the financial ecosystem as a heterogeneous graph of accounts, devices, merchants, IP addresses, and sessions, the system learns to detect abnormal network patterns such as fan-in/fan-out behavior, dense clusters, and transaction cycles that are characteristic of mule rings.

The project successfully integrates multi-source streaming data — including transactions, onboarding/KYC data, device telemetry, payment logs, and behavioral biometrics — into a unified representation. Through feature engineering, edge weighting, and GNN-based embedding generation, MuleGuard AI achieves an interpretable risk score for each event. A decision engine then maps this score into actionable categories (ALLOW, STEP-UP, HOLD, BLOCK), enabling real-time intervention.

Key contributions of this work include:

- A privacy-preserving architecture that allows cross-bank collaboration without sharing sensitive customer data.
- Explainable AI outputs, producing graph snapshots and top contributing features to support human analysts and regulatory audits.
- A modular, configuration-driven pipeline, making the solution portable and adaptable to varied banking environments.