CrediSure

PSB HACKATHON 2025

- Round 1

Fraud Detection & Defaulter Localization in Adversarial Financial Networks

Team:

CrediSure

Het Patel

Kanishk Gadiya

CrediSure

Table of content

CrediSure

Problem identification

- Fraudulent Activity Detection
  - Identify accounts or transactions that exhibit signs of fraud
  - Real-world fraud is subtle: patterns often resemble legitimate behavior
  - Data characteristics:
    - Noisy and obfuscated fields
    - Temporal irregularities
    - High class imbalance (few fraudulent cases)

- Last Known Location Estimation for Defaulters
  - Infer where known defaulters were last seen using tower logs
  - No direct geolocation — only indirect device-to-tower pings
  - Sparse, time-ordered connection logs with limited device metadata

Our Goal

- Build fraud detection and location-tracing models that are explainable, scalable, and robust — even in noisy, adversarial environments.

CrediSure

Understanding the data

🎯 Features

🔢 **Numerical** (Floats & Ints): Transaction volumes, frequencies, flags, ratios
🔤 **Categorical**: Encoded merchant types, anonymized user/device IDs
📅 **Datetime**: Timestamped logs for temporal modeling

📊 Dataset Snapshot

Rows: 377,241
Columns :139 features

📄 What the Data Represents

- Each row = A **snapshot of an account or transaction behavior**
- Combines multiple aspects:
- **User activity**: e.g., NUM_TXN_24HRS, TXN_AMT_MAX, ACCT_AGE
- **Behavioral indicators**: usage patterns, channel preference
- **System-observed flags**: potentially suspicious behavior or inconsistencies
- All personally identifiable details are **anonymized or obfuscated**

CrediSure

Challenges & Constraints

Noisy & Incomplete Data

- Random missing values in key fields
- Inconsistent time intervals in logs
- Sparse tower data and device activity

Intentional Obfuscation

- Merchant codes, user IDs, tower IDs all anonymized
- Makes feature interpretation and validation tricky

Adversarial Nature

- Fraudsters may mimic normal user behavior
- E.g., small, well-timed transactions to avoid detection

Temporal Complexity

- Sequence matters: timestamps affect both fraud patterns & movement
- Need to preserve chronological order in all modeling

Extreme Class Imbalance

- Fraud cases = rare
- Defaulters = limited subset
- Need anomaly detection, smart sampling, or cost-sensitive modeling

External Data Not Allowed

- No access to maps, coordinates, or external benchmarks
- Location estimation must rely only on tower IDs

CrediSure

Our Solutions

🕵️‍♂️ Fraud Detection

We aim to identify accounts or transactions that deviate from typical user behavior — even when fraudsters try to hide.

Key Ideas:

- Use a combination of:
  - **Anomaly detection models** (e.g., Isolation Forest)
  - **Supervised models** (e.g., Logistic Regression, LightGBM)
- Extract features from:
  - **Transaction behavior** (amounts, frequency)
  - **Temporal patterns** (spending times, gaps)
  - **Device-tower activity** (mobility inconsistencies)

Why it works:

- Flags outliers that mimic real users but have subtle pattern shifts
- Scalable + explainable via feature importance

📍 Defaulter Localization

Estimate the last known location of defaulters using tower logs.

Key Ideas:

- Trace device-to-tower logs by timestamp
-
- For each defaulter:
-
  - Identify the last valid tower connection
  -
  - Treat it as a proxy for last known location

Why it works:

- Intuitive and time-aware, No need for external location data
- Provides interpretable movement trails

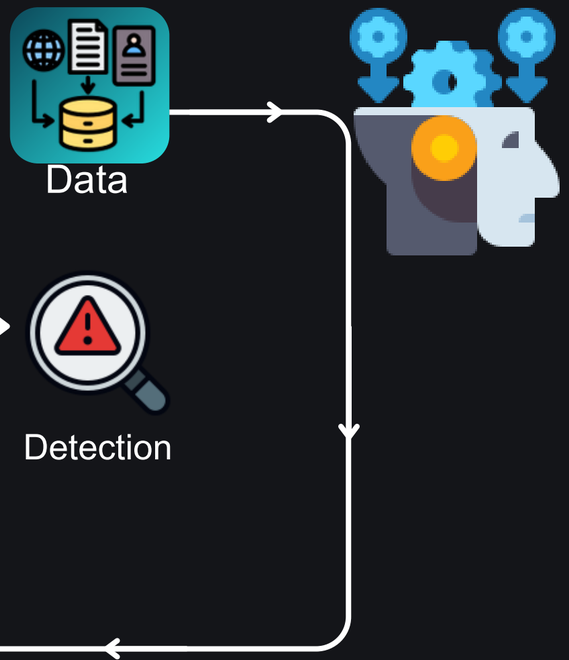
(Device + Time = Breadcrumb Path)

CrediSure

Methodology – Fraud Detection

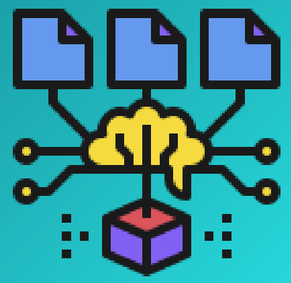
Interpretability

- Use of feature importance and SHAP values to explain model predictions
- Focus on transparency over black-box complexity



Modeling Approach

- Started with Isolation Forest for unsupervised anomaly detection
- Considering Logistic Regression or LightGBM for supervised modeling (if labels derived/generated)
- Output: Fraud score per account or transaction



Feature Engineering

Transaction Amount Patterns

- Daily, weekly, monthly spend summaries
- Max/min/spike detection
- Ratio of small vs large transactions

Device-Tower Correlation

- Device ID switching across towers
- Unusual location-based activity vs transaction origin
- Multiple accounts from the same device (possible fraud ring)

Velocity of Transactions

- Time gaps between consecutive transactions
- Sudden bursts or rapid-fire transactions
- Late-night or off-hour activity

CrediSure

Methodology – Defaulter Localization

Step-by-Step Process

Identify Defaulter Accounts
- Filter records to isolate accounts labeled or flagged as defaulters

Map to Device IDs
- Link each defaulter account to its associated device ID using provided logs

Sort Logs by Timestamp
- Arrange each device's tower connection records in chronological order

Extract Final Tower ID
- For each device, select the last valid tower connection
- Use this as a proxy for their last known location

Why This Works

Time-Aware Logic
- By sorting tower connections chronologically, we capture the most recent activity — allowing us to trace a defaulter's movement just before they went inactive.

Self-Contained & Compliant
- The method uses only the provided anonymized dataset, without relying on external tools like maps or coordinates — fully respecting the rules.

Transparent & Explainable
- The logic is easy to follow and explain: for every defaulter, we can point to the exact tower and timestamp of their last known connection — no black-box modeling involved.

Extension

- Apply clustering algorithms (e.g., K-Means, DBSCAN)
- Group nearby towers into broader zones or risk regions
- Helps generalize location without real coordinates

CrediSure

Tools & Technology

Python Ecosystem

- Pandas & NumPy – Data manipulation and preprocessing
- Scikit-learn – Machine learning models (e.g., Isolation Forest, Logistic Regression)
- Matplotlib & Seaborn – Basic visualizations and EDA

Version Control & Collaboration

- GitHub – Centralized codebase with:
  - Cleanly documented code
  - README with setup instructions

.

Collaborative workflow for seamless updates and tracking

Extra for Scaling

Prepared to scale up in Round 2 using:

- PySpark – For distributed data handling on large-scale operations
- Dask – Lightweight alternative for parallel computation in Python

Planned for Round 2

- SHAP / LIME – For interpretability and explanation of fraud predictions
- Streamlit / Dash – For building an interactive dashboard (fraud alerts & location insights)

CrediSure

Assumptions & Limitations

Device-to-Account Mapping is Reliable

- We assume that each device ID corresponds to a single user account, without duplication or shared usage.

Tower IDs Reflect Real-World Proximity

- Although geo-coordinates are hidden, we treat each tower ID as a valid proxy for the user's approximate location.

User Behavior is Consistently Captured

- The transaction and mobility patterns available in the logs are assumed to reflect real usage without major distortion.

Timestamps are Trustworthy

- All logs are assumed to follow correct chronological order, which is critical for both fraud detection and location tracing.

No Labeled Fraud Ground Truth

- Without actual fraud labels, we rely on unsupervised learning and cannot evaluate precision or recall directly in Round 1.

Anonymized and Sparse Features

- Many fields contain missing or deliberately obfuscated data, which limits interpretability and confidence in certain features.

Scalability Deferred to Future Rounds

- While methods are designed to scale, heavy data operations like full-batch feature generation or Spark processing were not implemented in this phase.

Feature Semantics are Ambiguous

- Obfuscated category or tower codes make it hard to validate assumptions about transaction types or regional clustering.

CrediSure

What's Next (If We Advance)

1. Scale with Spark or Dask

- Transition from sample PoC to full 377k+ row processing using PySpark or Dask
- Optimize for distributed feature engineering and parallel model training

2. Apply Deep Learning Techniques

- Use LSTM models for detecting complex fraud patterns over time (temporal modeling)
- Explore Graph Neural Networks (GNNs) to capture connections between accounts, devices, and tower hops

3. Tune for Adversarial Robustness

- Simulate mimicry attacks (e.g., low-value frauds) to test sensitivity
- Use cost-sensitive learning, robust metrics, and threshold tuning to minimize false negatives

4. Build an Interactive Dashboard

- Design a real-time fraud alert interface with key features and fraud scores
- Add a defaulter location tracker using inferred tower data
- Tools considered: Streamlit, Dash, or Plotly

CrediSure

Team Strength

Strong Foundation in ML & Data Analysis

We bring hands-on experience with machine learning, from preprocessing to model interpretation.

Comfort with Noisy, Imbalanced Data

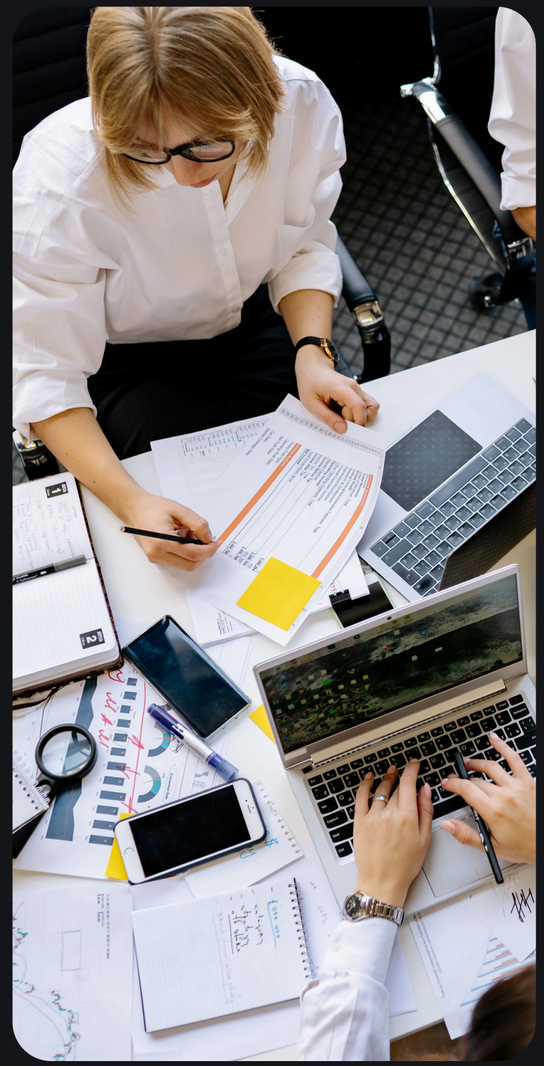
We've worked with real-world datasets before — messy, sparse, adversarial? We don't flinch.

Focused on Explainable AI

We value transparency and accountability. Our solutions prioritize interpretability alongside accuracy.

Domain Curiosity in Finance

We're passionate about applying data science to real financial problems — especially fraud detection and credit risk.



Our Vision

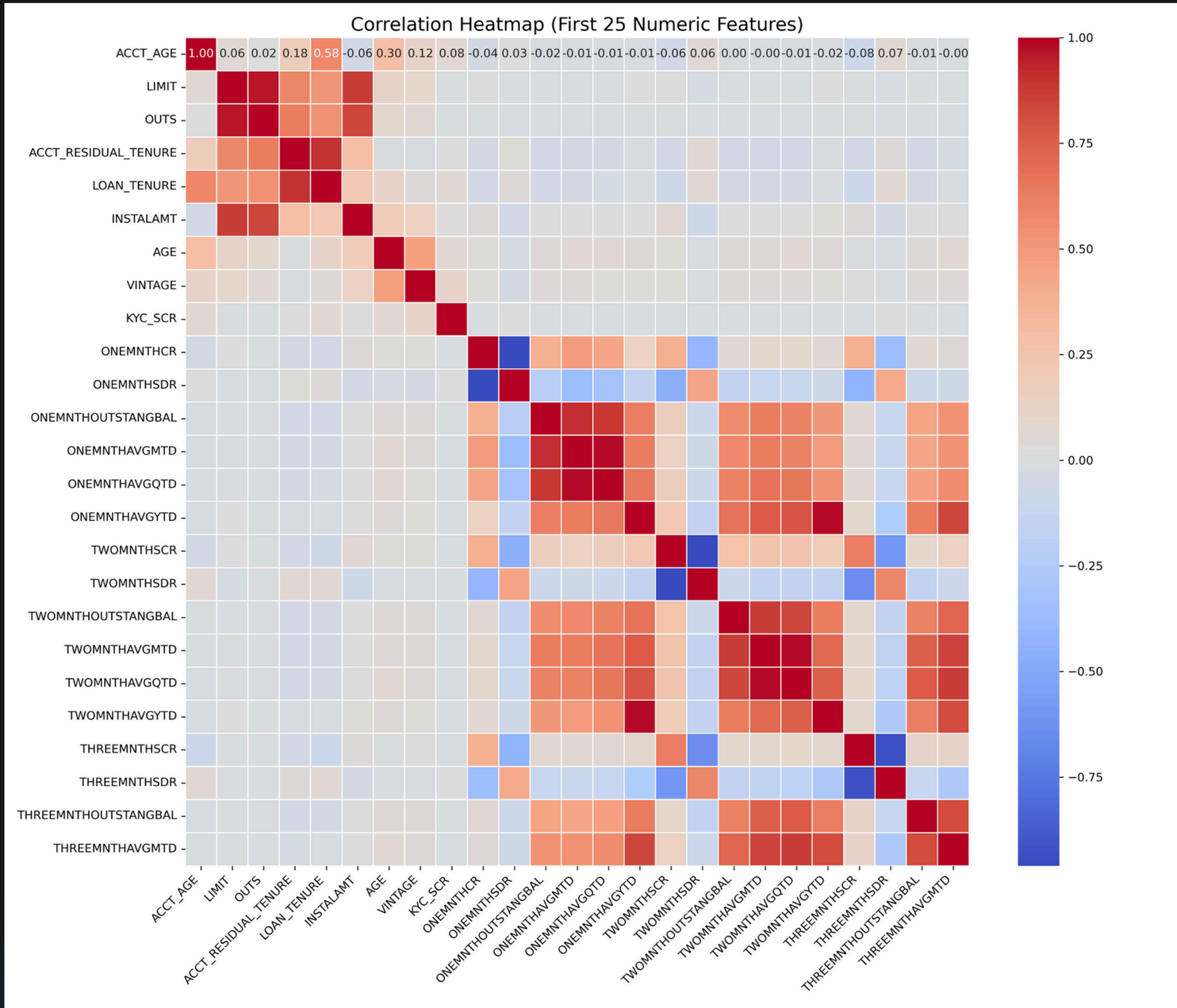
"To design scalable, ethical, and explainable data-driven systems that can thrive in high-stakes environments like finance."

- Whether it’s modeling human behavior or tracing digital footprints — we’re in it to solve, scale, and simplify.

CrediSure

Exploratory Graphs

Early Insights: Strong correlations in behavioral ratios & device movement metrics show promise for anomaly modeling.



CrediSure

Conclusion

Problem Recap

We tackled two critical challenges central to financial intelligence and digital security:

- Fraud Detection: Identifying anomalous behavior across anonymized transactional data
- Defaulter Localization: Estimating the last known location of users based on temporal tower connection logs

Key Takeaways

Focused on explainability, scalability, and real-world alignment

Built a solution designed to work even in noisy, incomplete, and adversarial environments

Laid out a clear roadmap for deeper modeling, including graph-based learning and live dashboards

Our Approach

- Applied a combination of unsupervised and interpretable machine learning techniques for fraud detection
- Developed a transparent, rule-based framework for tracing defaulters using only tower logs
- Ensured full compliance with all constraints, including the use of only provided, anonymized data

We envision systems that are not only data-driven — but resilient, explainable, and operational in India’s financial backbone.

CrediSure

Let’s not just detect risk — let’s outsmart it

Thank you

From Team:

CrediSure

HET PATEL p.vishnubhai@iitg.ac.in

KANISHK GADIYA g.kanishk@iitg.ac.in