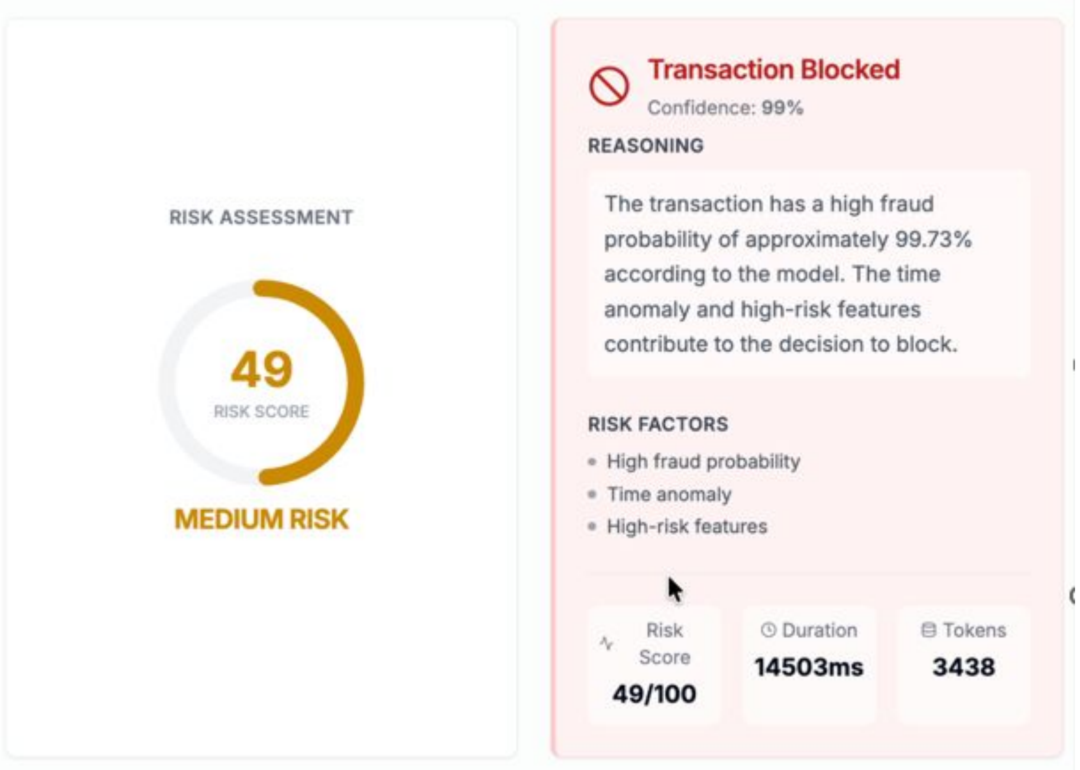
Agentic Fraud Detection System

A Machine Learning Approach to Financial Security



Term Project

Business Analytics for Managers

Students

Oğuzhan Kır & Fırat Olçum

Presentation Outline

Introduction & Motivation

The Challenge

- Financial fraud costs billions annually.
- Traditional systems lack adaptability and have high false positives.
- ML models need explainability for stakeholder trust.

Our Solution

- **XGBoost Model:** 99.76% AUC on temporal test data.
- **Agentic Framework:** ReAct-based reasoning for transparency
- **Real-time:** <50ms end-to-end inference latency

Exploratory Data Analysis

Dataset Overview

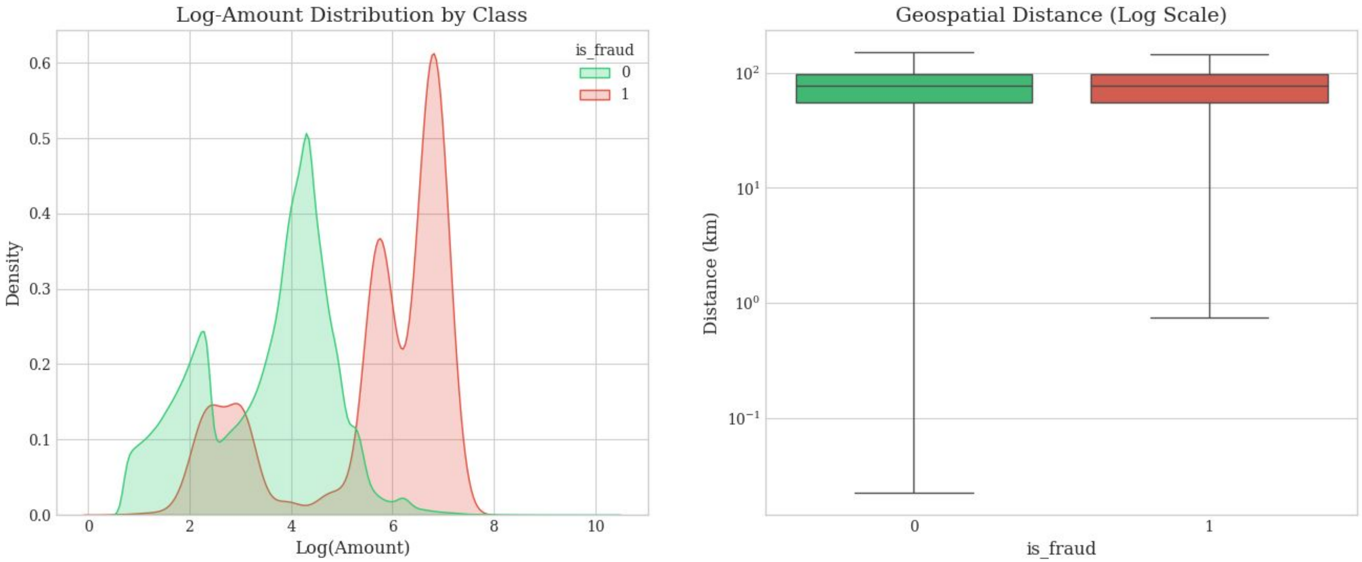
- 1,296,675 synthetic credit card transactions
- 7,506 fraudulent cases (0.58% prevalence)
- Temporal coverage: January 2019 – June 2020 (18 months)
- Features: Temporal, geospatial, transactional, demographic
- Designed to mirror real-world transaction distributions

Exploratory Data Analysis

Distributional Analysis

- Fraud shows bimodal distribution (low & high amounts).
- Log transformation improves class separation.
- Similar median distances but fraud shows higher variance.

Figure 1: Distributional differences in Amount and Location

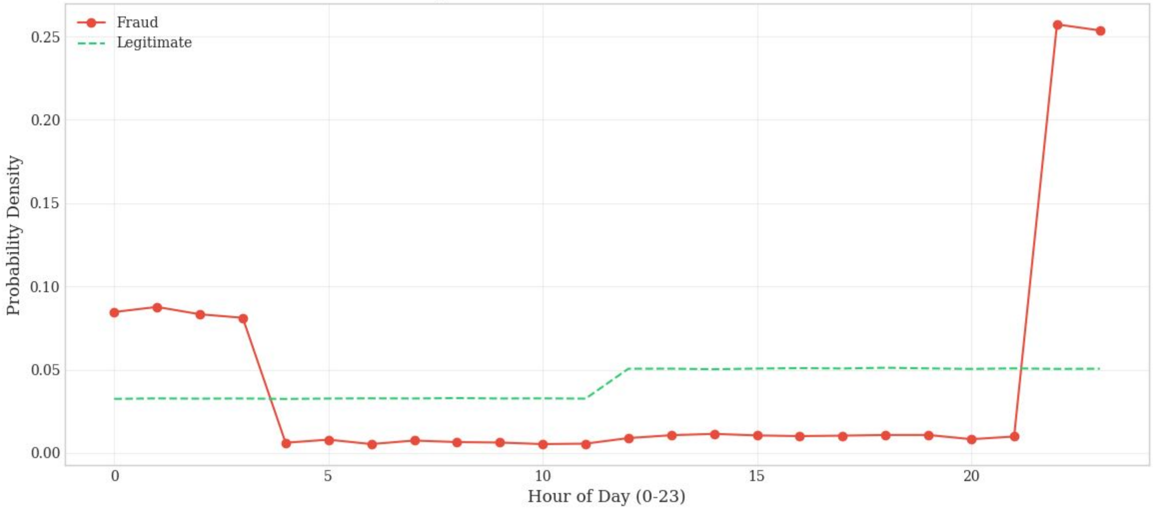


Exploratory Data Analysis

Temporal Pattern Recognition

- 25%+ of fraud occurs in 21:00-23:00 window.
- Legitimate transactions uniform (3-5%) across day.
- Near-zero fraud during business hours (05:00-20:00).
- Created hour_risk_score feature with elevated weights.

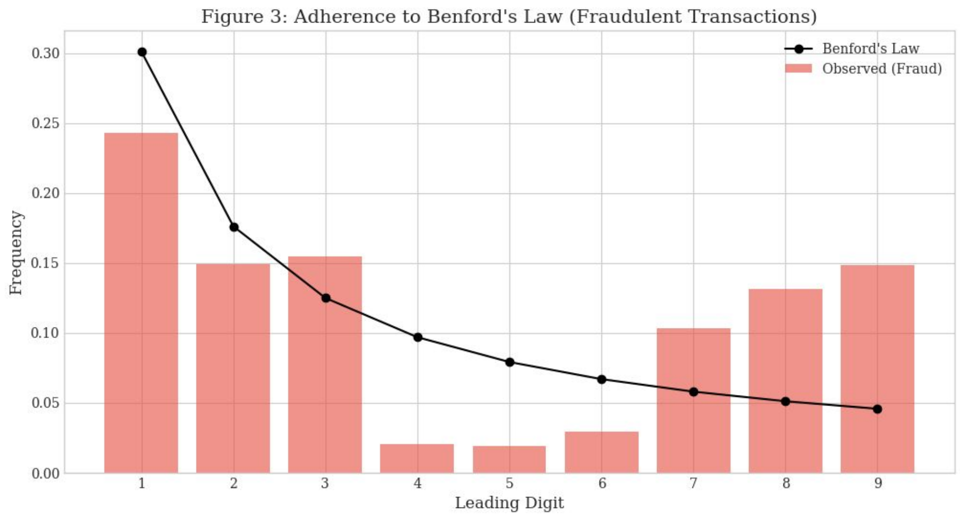
Figure 2: Diurnal Transaction Patterns



Exploratory Data Analysis

Benford’s Law Analysis

- Benford’s Law is that in natural datasets, leading digits follow logarithmic distribution. Deviation indicates fraud.
- Fraud shows strong deviation from expected.
- Over-representation of digits '1' (24%) and '9' (15%).
- Engineered benford_log_prob feature.



Feature Engineering

Feature Engineering Strategy

- **Temporal Features:**

  Cyclical Encoding**:** hour_sin, hour_cos, day_sin, day_cos.
- **Geospatial Analytics:**

  Haversine formula for geodesic distance.
- **Financial Profiling:**

  log_amt, relative amounts, Benford's Law probability.
- **Behavioral Aggregates:**

  Rolling windows (24h, 7d, 30d): transaction counts, amounts, frequencies.

Model Development

Model Training Strategy

- **Temporal Split (Forward Validation):**

  Training Set (80%): Jan 2019 – Mar 2020 (1,037,340 lines of data).

  Test Set (20%): Apr 2020 – Jun 2020 (259,335 lines of data).

  Prevents temporal leakage: Model trained on historical data only.

- **Nested Cross-Validation:**

  Outer Loop (5-fold): Performance estimation.

  Inner Loop (3-fold): Hyperparameter optimization with Optuna.

  100 trials per model using TPE sampler.

- **Candidate Models**

  XGBoost, LightGBM, Random Forest.

Results & Performance

Test Performance Results

- **XGBoost:** Precision 93.7%, Recall 82.5%
- **Random Forest:** Precision 94.4%, Recall 77.4%
- **LightGBM:** Precision 26.8%, 51.7% Recall

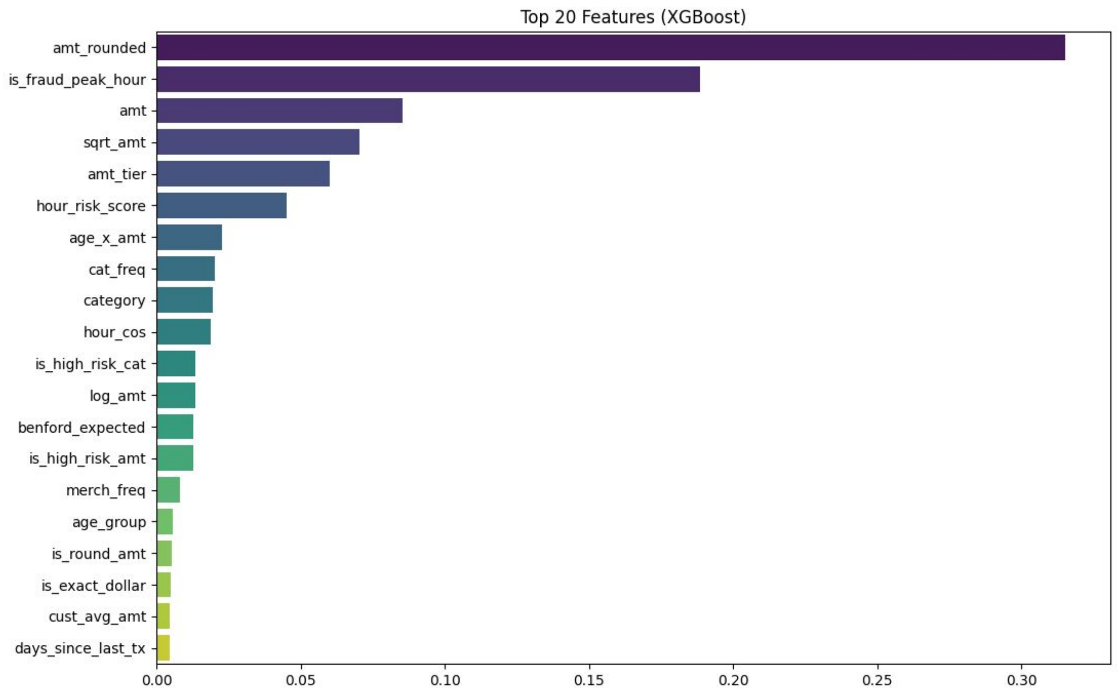
Winner: XGBoost with near-perfect class separation

Model	Recall	Precision	Real Cost
XGBoost	0.825098	0.937223	135350
LightGBM	0.517555	0.268647	392670
RandomForest	0.773732	0.943695	174710

Model Development

Top Features

- **amt_rounded (0.30)** : Round amounts
- **is_fraud_peak_hour (0.20) :** 21:00-23:00 flag
- **amt, sqrt_amt, amt_tier** : Amount variations
- **hour_risk_score (0.05)** : Temporal risk
- Amount features dominate top 6 positions



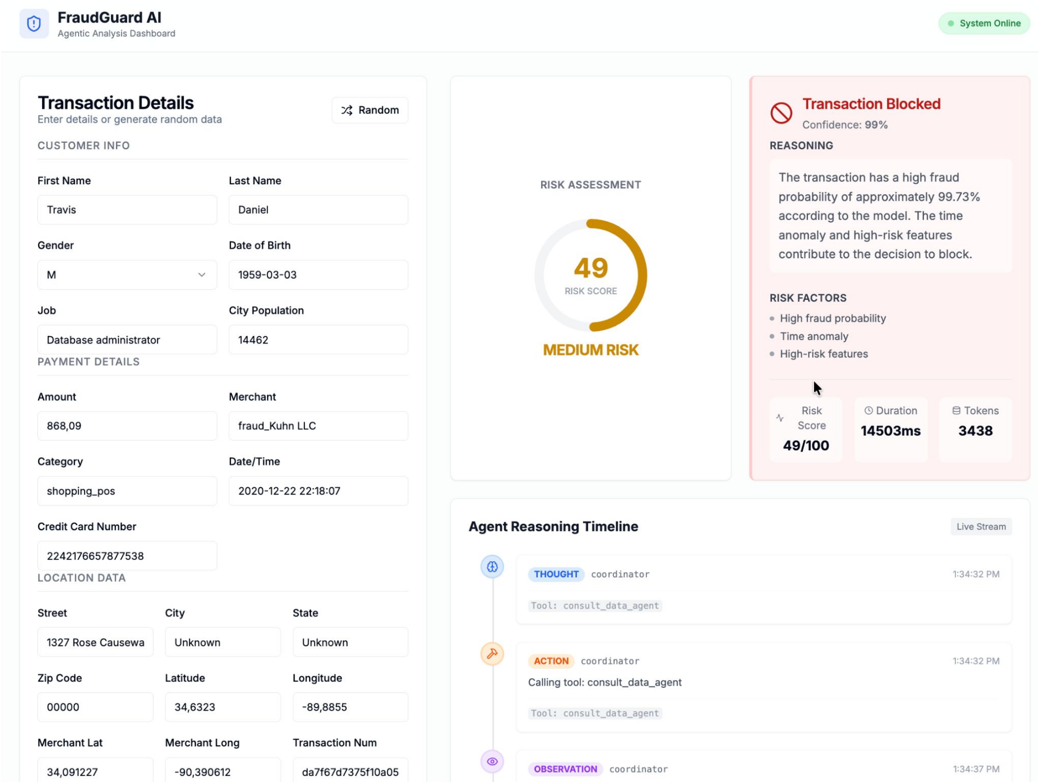
System Architecture

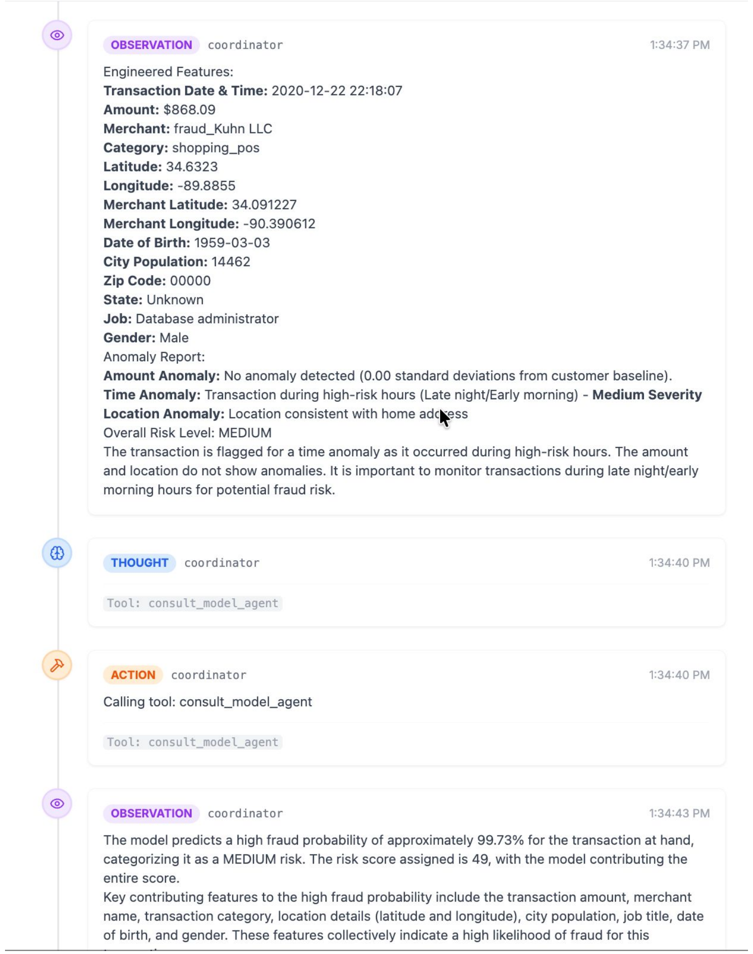
Agent Architecture & ReAct Workflow

- **Multi-Agent Hierarchy:**
  - **Coordinator Agent** : Decomposes tasks, delegates to sub-agents, final decision.
  - **Data Agent** : Uses detect_anomalies tool (Z-scores, Benford's Law, temporal risk).
  - **Model Agent** : Uses model_predictor and calculate_risk_score tools.
- **ReAct Loop Implementation:**
  - **THOUGHT** : Agent analyzes current state and plans next action.
  - **ACTION** : Agent calls specific tool (consult_data_agent, consult_model_agent).
  - **OBSERVATION** : Agent receives tool results and updates reasoning.
- **Tools & Capabilities:**
  - **detect_anomalies** : Statistical analysis, Benford's Law, temporal patterns.
  - **model_predictor** : XGBoost inference via ONNX Runtime (5ms).
  - **calculate_risk_score** : Combines probability + anomalies → 0-100 score.
  - **get_customer_profile:** Historical spending patterns and baselines.

System Architecture

Tought → Action → Observation





FraudGuard AI Dashboard: Transaction details, risk assessment, and decision

Business Impact

Cost-Benefit Analysis

- Baseline (no model): $1,501,200 in fraud losses.
- XGBoost Model: $908,220 net benefit.
- ROI: 60.5% reduction in fraud-related costs.

Operational Metrics

- 6,194 fraud cases detected and prevented.
- Only 416 false positives (6.3% false alarm rate).
- 1,312 fraud cases missed (17.5%).

Additional Benefits

- Real-time prevention at point-of-sale.
- Explainable decisions for regulatory compliance.
- Continuous improvement through analyst feedback.

Deployment & Technical Stack

Backend Stack

FastAPI

Asynchronous Python web framework

LangChain

Agent orchestration & LLM integration

ONNX Runtime

Optimized model inference (5ms latency)

WebSocket

Real-time streaming to frontend

Frontend Stack

Next.js

React 18 + TypeScript framework

CSS

Tailwind CSS + shadcn/ui components

RealTime

Real-time ReAct timeline visualization

Docker Deployment

Compose

Multi-container setup with Docker Compose

Backend

Python 3.11-slim (512MB memory limit)

Frontend

Node 20-alpine (256MB memory limit)

Monitoring

Production ready with health monitoring

Conclusion & Future Work

Key Achievements

- State-of-the-art performance: ROC-AUC 0.9976.
- Production ready: <50ms latency, containerized deployment.
- Explainable AI: Multi-agent architecture bridges accuracy and trust.
- Business impact: $908K annual savings (60.5% ROI).

Future Enhancements

- Federated Learning: Train on decentralized data while preserving privacy.
- Graph Neural Networks: Detect collusion rings and mule accounts.
- Reinforcement Learning: Optimize decision thresholds based on outcomes.
- Online Learning: Continuous adaptation to emerging fraud tactics.

Recommendations

- Deploy XGBoost with agentic explainability as primary system.
- Implement monthly retraining with sliding temporal window.

References

[1] Dataset: Cartella, F. (2018). Simulated Credit Card Transactions Dataset. Kaggle. https://www.kaggle.com/datasets/kartik2112/fraud-detection

[2] XGBoost: Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[3] Benford's Law: Durtschi, C., Hillison, W., & Pacini, C. (2004). The Effective Use of Benford's Law to Assist in Detecting Fraud in Accounting Data. Journal of Forensic Accounting, 5(1), 17-34.

[4] Imbalanced Learning: He, H., & Garcia, E. A. (2009). Learning from Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263-1284.

[5] ReAct Framework: Yao, S., et al. (2023). ReAct: Synergizing Reasoning and Acting in Language Models. International Conference on Learning Representations.

[6] Temporal Validation: Cerqueira, V., Torgo, L., & Mozetič, I. (2020). Evaluating Time Series Forecasting Models: An Empirical Study on Performance Estimation Methods. Machine Learning, 109(11), 1997-2028.

[7] Explainable AI: Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[8] Feature Engineering: Zheng, A., & Casari, A. (2018). Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists. O'Reilly Media.