SketchGNN — Streaming Graph Learning for Real-Time Mule Fraud Detection

This talk presents a production-ready approach that compresses dense transaction graphs into constant-size streaming summaries for millisecond fraud detection at scale.

TEAM NAME - PATTERN PATROL

SHASHATH D - 23MIM10027

DERIN DENNY MATHEW - 23MIM10065

DISOOR S -23MIM10121

PREM RAMAMOORTHY - 23MIP10019

The Problem —Fraud Moves Faster Than Detection

Speed of Attack

Fraud networks move funds within minutes; cash-out often completes before traditional detection triggers.

Extreme Density

Transaction graphs are hyperconnected — some accounts contact 50k+ entities/day, producing enormous adjacency volume.

Latency Requirements

Real-time detection must operate in milliseconds. Any neighbor expansion can break SLAs.

Traditional Expansion Fails

Graph neighbor expansion introduces latency spikes and systemic instability under bursts.

Why Traditional GNNs Fail at Scale

- Neighbor expansion + multi-layer message passing requires fetching large adjacency sets.
- Real-time scoring becomes unstable during traffic spikes due to recursive neighborhood expansion.
- Caching neighbors is ineffective under rapidly changing transaction streams.
- Memory explodes when storing multi-hop features for millions of active nodes.
- Inference latency rises nonlinearly — system fails under realistic fraud traffic.

This is the "Neighborhood Explosion Problem" — the central scalability bottleneck for graph ML in streaming fraud detection.

Core Idea — SketchGNN

SketchGNN enables real-time fraud detection by replacing expensive neighbor searches with a small, continuously updated behavioral summary stored at each account. As transactions arrive, lightweight streaming updates capture interaction frequency, diversity of connections, and sudden activity changes. These summaries approximate local graph behavior without traversing large transaction networks, avoiding scalability and latency issues. As a result, the system delivers fast, stable fraud detection at million-scale using minimal computational resources.

Frequency Sketches

Approximate counts for repeated counterparties, ATM usage, device reuse — updated per event.

Distinct Counters

Estimate unique counterparties (HyperLogLog) to detect funnels and distributors without enumerating edges.

Velocity Metrics

EWMA captures burstiness and rapid cash-out patterns for timeline-aware scoring.

Algorithms — Intuitive, Lightweight, Robust

Count-Min Sketch

It efficiently estimates how frequently an account interacts with specific counterparties without storing full transaction records. This helps detect suspicious repetition patterns such as repeated transfers, shared ATMs, or reused devices using minimal memory.

HyperLogLog

HyperLogLog estimates the number of unique senders or receivers connected to an account using compact probabilistic counting. This enables detection of funnel or distributor behavior without maintaining large adjacency lists.

Exponentially Weighted Moving Average

EWMA tracks how transaction activity changes over time by giving more importance to recent events. This allows the system to quickly identify sudden spikes or bursts that often indicate fraud escalation.

Risk Model

A lightweight classifier uses the sketch-generated features to compute a fraud risk score in real time. More computationally expensive analysis is triggered only for high-risk cases, keeping the system efficient and scalable.

Each algorithm is chosen for constant memory footprint and constant-time update behavior — ideal for per-node streaming state.

Two-Speed Risk Engine — Efficient Layering

Layer 1 — Fast Path

Lightweight sketch features + threshold rules. Millisecond decisions for the majority of events. Minimal compute, broad coverage.

Layer 2 — Deep Path

Triggered for high-risk entities. Runs compressed graph reasoning, richer models, or human review. Limits expensive computations to a tiny fraction of traffic.

This staged approach reduces average cost per transaction and keeps system latency stable under burst traffic.

Unified Entity Graph ER Model (Accounts–Wallets–Devices–ATMs)

High-Level Real-Time Mule Detection System (Cross-Channel, Graph + GNN)

Real-Time Streaming Fraud Detection Pipeline

- Transaction Ingestion: Incoming transactions are streamed through Kafka, ensuring reliable and high-throughput data collection in real time.
- Stateful Stream Processing: Apache Flink processes each event and updates per-node behavioral sketches while maintaining state consistency.
- Efficient Storage: Sketch states are backed by systems like RocksDB or Redis to provide fast access and fault tolerance.
- Real-Time Risk Scoring: An online scoring model evaluates updated features instantly and generates a fraud risk score within milliseconds.
- Decision Execution: A decision engine applies predefined policies to automatically Allow, Hold, or Block transactions based on the risk score.

Edge Detection Without Traversal

Funnel Detection

High incoming distinct senders with low outgoing distinct receivers → candidate mule aggregator.

Distributor Signals

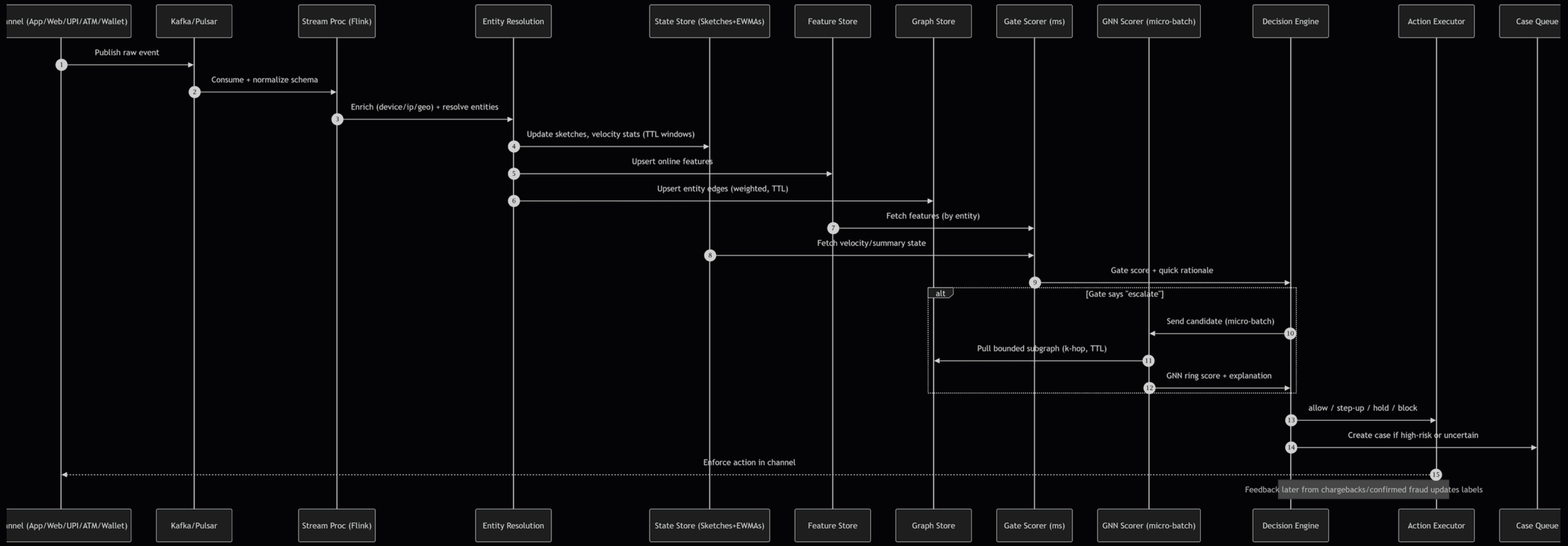
Low inbound distinct count but spikes in outgoing frequency sketch entries indicate dispersal nodes.

Rapid Receive→Send Patterns

EWMA velocity spikes plus repeat ATM/device counts expose fast cash-out chains.

Combined sketch features approximate local topology and temporal behavior well enough to surface mule rings without expanding neighbors.

Real-Time Transaction Scoring Sequence (Two-Speed: Gate + GNN)



Real-Time Transaction Scoring Sequence (Two-Speed: Gate + GNN)

Scalability & Efficiency — Key Guarantees

Constant Memory

Fixed-size sketch per node — memory scales linearly with active nodes, not edges.

Constant Update Time

O(1) updates per transaction — suitable for high throughput and bursty arrivals.

No Graph Expansion

Avoids adjacency fetches and DB bottlenecks — stable millisecond inference under load.

Production Scale

Designed for 1M+ transactions/hour with predictable resource usage and graceful degradation.

Impact & Innovation

SketchGNN enables practical graph intelligence on streaming financial data by trading exact adjacency for provable, compact summaries. The result: real-time mule fraud detection at massive scale without graph DB bottlenecks.

Systems + ML

Combines distributed streaming, probabilistic data structures, and lightweight ML for robust production performance.

Operational Ready

Works with Kafka/Flink/RocksDB/Redis and slots into existing pipelines with minimal overhead.

Conclusion

SketchGNN enables big graph intelligence with tiny memory. It delivers stable, millisecond fraud detection where classical GNNs cannot.

THANK YOU