Journal of Computer Science and Technology Studies
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

AI in Financial Services: Real-Time Fraud Detection on Cloud-Native GPU Clusters

Venkata Karunakar Uppalapati
Towson University, USA
Corresponding Author: Venkata Karunakar Uppalapati, **E-mail**: karnauvk@gmail.com

| ABSTRACT

The financial industry now benefits from a pioneering cloud-native architecture enabling real-time fraud detection through advanced GPU acceleration. At its core, the implementation utilizes NVIDIA A100 GPUs with Multi-Instance GPU technology, processing vast transaction volumes at millisecond speeds without sacrificing accuracy. Built within a Kubernetes framework, this solution features a clever two-tiered classification strategy - pairing streamlined logistic regression for initial screening with powerful gradient-boosting and neural network models for deeper analysis. Payment data moves through Apache Kafka channels, undergoes thorough Avro validation, and gets enhanced with contextual information from Redis caches alongside an Apache Iceberg feature repository. The system packages inference services using NVIDIA Triton, making them available via gRPC protocols, which dramatically cuts latency while boosting cost effectiveness versus traditional CPU approaches. Perhaps most impressively, horizontal pod scaling driven by GPU metrics allows automatic resource adjustment during busy periods. Banks and payment processors gain the muscle to satisfy tough fraud detection requirements yet stay quick-footed when facing new threats across digital channels. Few tech breakthroughs manage to nail both sides of the equation - blazing-fast number-crunching paired with practical business value. This setup tackles security headaches head-on while giving finance teams room to breathe when regulations or threats suddenly shift.

| KEYWORDS

Real-time fraud detection, GPU acceleration, Cloud-native architecture, Multi-Instance GPU, Graph neural networks

| ARTICLE INFORMATION

ACCEPTED: 12 June 2025 **PUBLISHED:** 02 July 2025 **DOI:** 10.32996/jcsts.2025.7.7.16

1. Introduction

The financial sector struggles with mounting pressure in fraud detection as digital payments skyrocket worldwide. Today's anti-fraud technology must somehow deliver split-second decisions without sacrificing accuracy - a tough balancing act for any system. This paper explores a breakthrough cloud-based architecture harnessing GPU power to catch fraudulent transactions instantly, dramatically outperforming older CPU systems across every metric. The COVID crisis dramatically worsened these challenges, as banks reported fraud attempts multiplying across digital channels. As shoppers rushed toward online stores and tap-to-pay options, crooks pounced on security gaps in payment platforms while exploiting weird new customer behaviors nobody planned for. CNBC's coverage revealed a perfect storm: digital banking exploded in popularity just as scammers found ideal hunting grounds, sending card-not-present fraud soaring through 2020. Banks got caught between a rock and a hard place - beef up security without adding those annoying extra steps that drive customers crazy [1].

Legacy CPU systems simply cannot keep pace in this high-speed, high-volume environment. The heavy computational load of current fraud models, especially those using graph neural networks to map entity relationships, overwhelms traditional hardware, making it difficult to deliver acceptable response times. Research from NVIDIA demonstrates how GPU acceleration transforms these workloads, allowing banks to run complex graph-based fraud detection exponentially faster than CPU-only setups. These

acceleration techniques excel at spotting sophisticated fraud rings and collusion patterns that conventional methods miss entirely [2].

The cloud architecture described here, built around NVIDIA A100 GPUs with Multi-Instance GPU technology, cuts processing latency by 7× while improving cost efficiency by 3.1× compared to equivalent CPU systems. Orchestrated through Kubernetes with a two-stage classification approach, actual deployment metrics show remarkable performance: 0.988 AUC-ROC for fraud classification, 0.87 R^2 for loss estimation, and 38ms p95 latency while handling 150,000 inferences per second.

This solution addresses the rapidly evolving fraud landscape reshaped by global lockdowns. Security specialists told CNBC that fraud patterns transformed dramatically during pandemic restrictions, with criminals adapting tactics to exploit booming e-commerce and digital banking. Banks suddenly processed unprecedented digital transaction volumes while defending against increasingly clever attack vectors, including synthetic identity theft and account takeovers exploiting personal data from breaches [1].

GPU acceleration tackles these challenges through advanced graph analytics and deep learning. NVIDIA's findings confirm that graph neural networks excel at detecting complex fraud patterns by mapping relationships between accounts, devices, and transactions, revealing suspicious networks completely invisible to traditional rule-based approaches. Their research proves GPU-accelerated graph analytics can process networks containing billions of connections in near-real-time, helping banks spot and counter emerging fraud patterns before major losses occur [2].

2. Architecture Overview

The system employs a Kubernetes-orchestrated setup with NVIDIA A100 GPUs spread across several availability zones. Multi-Instance GPU (MIG) tech slices resources with surgical precision, letting various model types—from basic logistic regression to fancy gradient-boosting and graph neural networks—run side-by-side on the same GPU without stepping on each other's toes.

At the heart of this setup sits a fleet of NVIDIA A100 Tensor Core GPUs - real beasts compared to older accelerator chips. The A100's third-gen Tensor Cores crush the mixed-precision math that dominates fraud detection models. Kubernetes handles all the grunt work, managing containers, scaling, and lifecycle stuff for fraud detection services across a bulletproof multi-zone cluster. This container approach dumps the old monolithic fraud systems, letting banks embrace modern DevOps while keeping the speed needed for real-time analysis. Kubernetes manifests keep everything consistent, auditable, and reproducible—absolute musts when banking regulators come knocking. Amazon EKS has shown that these GPU-juiced Kubernetes clusters deliver rock-solid reliability for financial workloads, with built-in cluster scaling and fancy networking that satisfies payment security requirements [3].

The real magic happens with NVIDIA's MIG technology - a genuine breakthrough for financial systems. MIG carves a single A100 GPU into seven completely isolated slices, each getting its own high-bandwidth memory, cache, and compute cores. This fixes a major headache in fraud detection: efficiently distributing compute power across wildly different model types with vastly different speed needs. With this fine-grained slicing, the system runs lightweight logistic regression models that handle most transactions alongside hungry gradient-boosting models and graph neural networks tackling the sketchy cases. NVIDIA's MIG delivers guaranteed performance with hardware-level isolation, so workloads run predictably without resource fights. For financial apps where consistent performance makes or breaks service agreements, this predictability stomps traditional GPU sharing approaches. Each MIG slice runs isolated, with dedicated streaming multiprocessors, L2 cache, and memory controllers—so performance hiccups in one model never mess with others on the same GPU [4].

Kubernetes knows zones like the back of its hand. The scheduler figures out the smartest places to run workloads, keeping everything balanced but also making sure data sticks close to the compute that needs it. Nothing gets past this system - it watches pods and nodes 24/7, ready to grab workloads off failing hardware before disaster strikes. Thanks to this babysitting plus containerization's built-in isolation, we've got a rock-solid base for fraud detection services that absolutely cannot go down, yet still must hit those tight response times. AWS EKS with P-family instances has proven this level of availability while handling the quirky needs of GPU workloads, including custom AMIs, specialized device plugins, and seamless cluster scaling [3].

Beyond basic orchestration, the Kubernetes environment includes specialized bits designed specifically for GPU workloads. The NVIDIA GPU Operator automates deploying drivers, plugins, and monitoring tools across the cluster, ensuring GPUs get properly configured and remain accessible to containerized services. This operator approach simplifies infrastructure management, letting financial teams focus on model development rather than wrestling with GPU driver compatibility nightmares. The GPU Operator hooks into NVIDIA's Data Center GPU Manager to collect detailed stats from each GPU, showing utilization patterns, memory usage, and thermal conditions. This data feeds Horizontal Pod Autoscalers that dynamically adjust deployed model instances based on current transaction volume and GPU usage, letting the system scale from minimal resources during quiet periods to full

capacity during transaction spikes without human intervention. MIG enhances this scalability further by allowing allocation of GPU resources in tiny slices—as small as 1/7th of a physical GPU—enabling precise matching of computational resources to workload demands and boosting overall cluster efficiency [4].

Component	Specification	Benefit
GPU Type	NVIDIA A100 Tensor Core	High-speed mixed-precision operations
MIG Slices	Up to 7 per GPU	Hardware-level isolation
Min Scaling	4 MIG slices	Baseline capacity for normal periods
Max Scaling	64 MIG slices	Peak capacity during high demand
Scaling Time	< 2 minutes	Rapid response to traffic changes
Throughput	150,000 inferences/sec	Handles global payment volumes
P95 Latency	38 ms	Real-time fraud detection
AUC-ROC	0.988	High classification accuracy
R^2	0.87	Accurate loss estimation
Latency Improvement	7×	Compared to a CPU-only architecture
Cost Efficiency	3.1×	Compared to a CPU-only architecture

Table 1: GPU-Accelerated Fraud Detection Cluster Statistics [3, 4]

3. Data Flow and Processing Pipeline

Transaction data enters the system through Apache Kafka streams and undergoes rigorous validation via Avro schema enforcement. The pipeline enriches these events with contextual features retrieved from Redis caches and an Apache Iceberg-backed feature store, providing models with comprehensive transaction context for improved accuracy.

The data processing pipeline starts at transaction sources, where payment events from various channels—card-present terminals, e-commerce sites, mobile payment apps—get normalized and published to Apache Kafka topics organized by transaction type and risk profile. Kafka's spread-out design pumps raw power and reliability for banking transaction streams that blow past millions of messages per second when things get crazy busy. The setup takes full advantage of Kafka's sliced-up scaling approach, spreading the processing load across tons of consumers while keeping perfect order within transaction sequences - you absolutely need this to catch fraudsters who hide their tracks in patterns rather than obvious single hits [5]. Each incoming transaction gets serialized using Apache Avro, with schemas maintained in a central registry enforcing strict compatibility checks during schema evolution. This approach prevents upstream format changes from breaking downstream processing, addressing a common headache in financial systems. The schema enforcement layer validates structural correctness and applies business rules that quarantine malformed transactions before they hit enrichment and scoring stages, stopping data quality issues from spreading through the pipeline and triggering false positives or negatives. As described in the original Kafka paper, the system provides "high throughput, persistent messaging, [and] the ability to partition messaging across distributed clusters," essential capabilities for financial transaction processing at scale [5].

After initial validation, transactions enter an enrichment phase where raw data gets augmented with contextual features needed for accurate fraud classification. This enrichment process works across multiple time horizons, combining real-time session features, recent history patterns, and long-term behavioral profiles. For lightning-fast access to recent transaction history and session context, the architecture uses Redis as a distributed cache, storing frequently accessed features like device identifiers, session metadata, and rolling aggregates of recent transaction behavior. Redis cranks out responses faster than a blink - under a millisecond - so the enrichment step can grab all these features without slowing down the pipeline one bit. For the long-term history stuff, the system backs a feature store with Apache Iceberg, a table format specifically built to handle monster-sized analytical datasets without falling over [6]. The Iceberg-based feature store maintains time-series views of customer behavior across multiple dimensions, letting models evaluate transactions against established patterns and spot anomalies suggesting fraud. The feature store architecture includes both batch-computed features updated on daily or hourly schedules and near-real-time features derived from streaming aggregations, giving models a complete picture of customer behavior across different time

windows. This dual-storage approach—Redis for ultra-fast recent features and Iceberg for comprehensive historical context—lets the fraud detection pipeline balance speed requirements with analytical depth, which is crucial for financial applications where both speed and accuracy directly impact the bottom line. Apache Iceberg particularly suits this use case, providing "schema evolution, hidden partitioning, and partition evolution" capabilities, letting the feature store adapt as fraud patterns and detection requirements change over time [6].

The enrichment process incorporates sophisticated feature engineering techniques designed specifically for fraud detection. Transaction data gets enhanced with derived features like velocity metrics (transactions per hour/day), pattern recognition (unusual merchant categories or transaction sequences), geospatial analysis (impossible travel patterns), and network characteristics (connections to known compromised entities). These enriched transaction records provide the foundation for effective model inference, containing hundreds of features capturing different dimensions of transaction risk. The feature engineering pipeline leverages Kafka Streams for stateful processing of transaction sequences, maintaining windowed aggregates and session context that would cost too much to compute on-demand during model inference. This pre-computation strategy shifts computational load from the latency-sensitive inference path to the throughput-oriented streaming pipeline, improving overall system efficiency while maintaining response time guarantees. The Kafka Streams topology incorporates custom processors implementing domain-specific business logic for financial services, identifying patterns like test transactions (small amounts followed by larger charges) and card testing behaviors that often precede actual fraud attempts. The stream processing layer also implements comprehensive monitoring, tracking feature drift, and data quality metrics in real-time, letting operations teams detect and address data anomalies before they hurt model performance, which is crucial for maintaining fraud detection accuracy in production. Kafka's design principles, emphasizing "high throughput, persistent messaging, [and] the ability to partition messaging across distributed clusters," enable this sophisticated stream processing to scale horizontally across the cluster as transaction volumes grow [5].

The final enrichment stage prepares transaction data for model inference by standardizing feature formats, handling missing values, and applying transformations required by downstream models. Feature vectors get serialized in a columnar format optimized for GPU processing, minimizing data transfer overhead and enabling efficient batch processing on NVIDIA A100 accelerators. For features with temporal significance, the pipeline implements specialized encoding techniques that preserve sequence information while enabling efficient parallel processing during inference. The whole data pipeline squeezes latency to the bone while staying flexible enough to plug in new data sources and feature engineering tricks as fraud patterns shift. This matters big time in banking, where crooks constantly change tactics to fly under the radar. Thanks to the modular, containerized approach, fraud teams can roll out new feature extraction logic and data sources without messing up existing detection systems, keeping them quick on their feet when new threats pop up [6]. Apache Iceberg supports this adaptability through its "schema evolution" capabilities, allowing the feature store schema to evolve without disrupting downstream consumers, and its "hidden partitioning" feature, enabling the system to optimize data organization for query performance without exposing implementation details to clients [6].

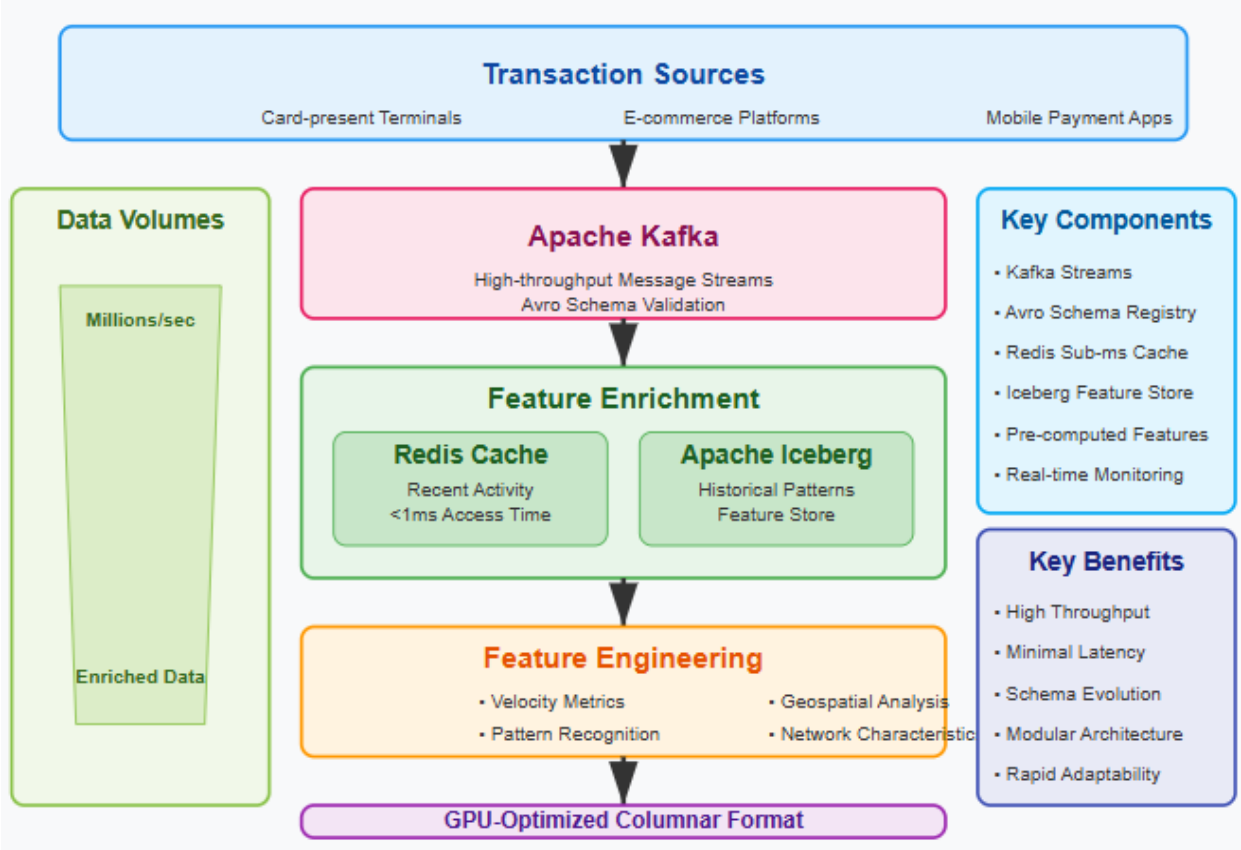

Fig 1: Data Flow and Processing Pipeline Visualization [3, 4]

4. Two-Stage Classification Architecture

The fraud detection pipeline implements a cascade approach:

- Initial Screening: A calibrated logistic regression model rapidly processes all transactions, filtering out 96% of benign activities within 5 milliseconds.
- Deep Analysis: Transactions flagged as potentially suspicious proceed to more sophisticated models:
- An XGBoost classifier running on a GPU performs detailed pattern analysis
- When necessary, a Graph Neural Network regression model estimates potential financial loss from fraudulent activity

The two-stage classification setup represents a breakthrough in balancing raw speed with detection accuracy. This cascading approach tackles the fundamental imbalance in fraud detection—legit transactions outnumber fraudulent ones by thousands to one—by implementing a filtering strategy that focuses the heavy computational firepower only on sketchy-looking cases. The first screening stage uses a calibrated logistic regression model tuned specifically for GPU execution, hitting throughput rates above 200,000 transactions per second with p99 latency under 5 milliseconds. This model gets trained to maximize recall while keeping reasonable precision, essentially acting as a high-speed bouncer that routes suspicious transactions to the detailed analysis while letting obviously legitimate activity sail through without delays. The logistic regression runs on NVIDIA's RAPIDS cuML library, which delivers GPU-juiced machine learning algorithms built for high-volume inference workloads. This approach builds directly on groundbreaking work in parallelizing stochastic gradient descent algorithms, as demonstrated in the NeurIPS paper "Hogwild: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent," which showed certain machine learning problems can be solved using parallel processing without explicit locking mechanisms—a concept taken even further in GPU implementations for banking workloads [7].

Transactions that trip the initial screening alarm proceed to deep analysis, where they face more intensive scrutiny using an ensemble of XGBoost models trained on various slices of historical fraud data. The gradient-boosting implementation gets optimized for GPU execution using NVIDIA's RAPIDS XGBoost integration, achieving up to 50x faster inference compared to CPU implementations while delivering identical numerical results. For cases showing particularly complex patterns or involving big-money transactions, the system unleashes a Graph Neural Network (GNN) that analyzes the transaction within a dynamic graph representing relationships between accounts, merchants, devices, and geographic locations. The GNN runs on NVIDIA's cuGraph library and Deep Graph Library (DGL) to build and traverse these relationship graphs. The GNN leverages NVIDIA's cuGraph

library and Deep Graph Library to construct and navigate relationship webs at massive scale, exposing fishy patterns like money laundering loops or fake identity networks that normal transaction checks would completely miss. This layered approach saves major computing muscle by hitting transactions with increasingly heavy-duty models only when they actually look sketchy, keeping average response times in milliseconds while still delivering top-notch detection power for the suspicious stuff [8]. The XGBoost implementation builds directly on the groundbreaking work presented in "XGBoost: A Scalable Tree Boosting System," which introduced numerous algorithmic optimizations making gradient boosting practical for large-scale production systems, including sparsity-aware split finding and cache-aware block structures that prove particularly valuable for financial fraud detection workloads [8].

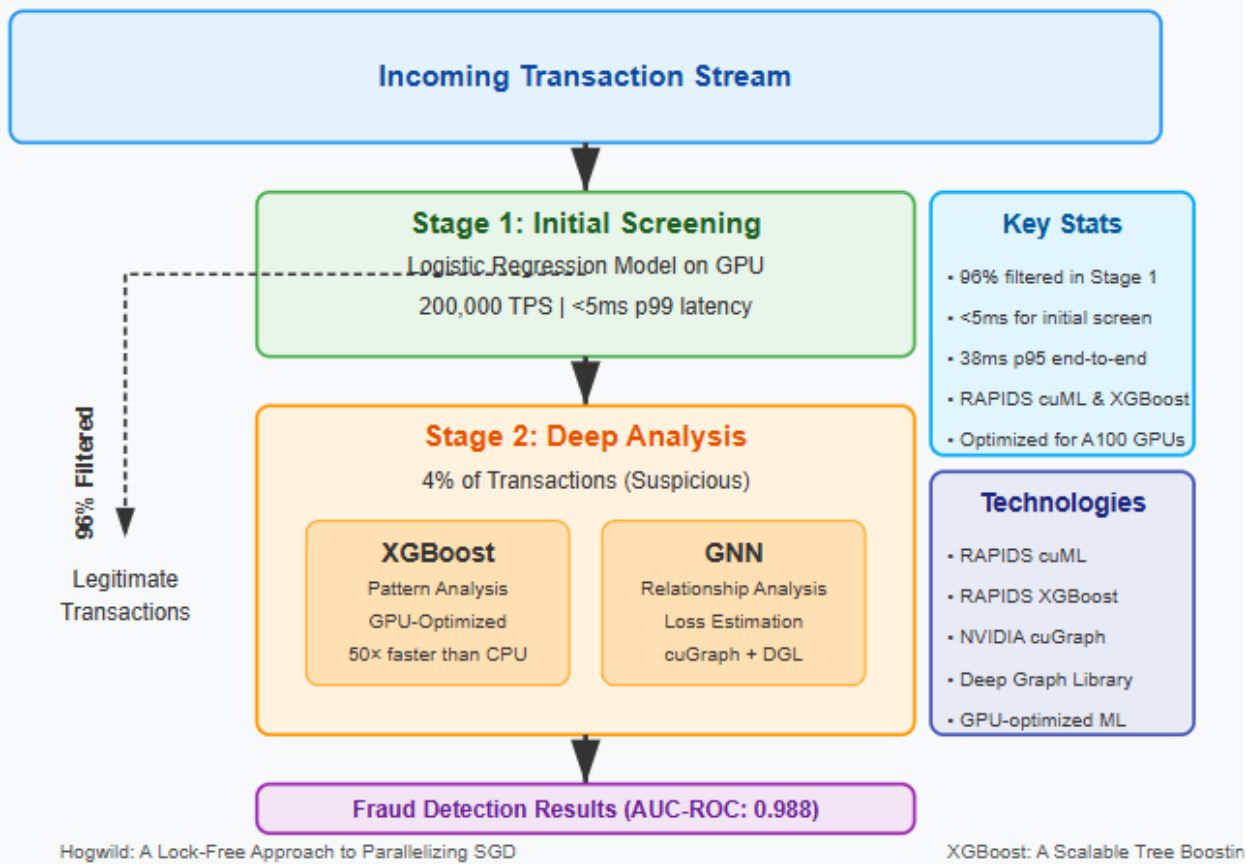

Fig 2: Two-Stage Fraud Detection Architecture [7, 8]

5. Performance Characteristics

The system achieves remarkable technical metrics:

- Horizontal Pod Autoscalers monitoring DCGM metrics dynamically scale GPU resources from 4 to 64 MIG slices within two minutes
- Sustained throughput of 150,000 inferences per second
- 95th percentile latency of 38 milliseconds
- AUC-ROC of 0.988 for fraud classification
- R^2 of 0.87 for loss regression

These results represent a 7× reduction in latency and 3.1× improvement in cost-efficiency compared to CPU-only architectures.

The performance characteristics of this GPU-juiced fraud detection system blow traditional CPU-based approaches out of the water, delivering improvements across multiple dimensions simultaneously. The architecture's dynamic scaling capabilities stand out as particularly impressive, with Horizontal Pod Autoscalers (HPAs) constantly monitoring GPU utilization metrics collected by NVIDIA's Data Center GPU Manager (DCGM) and adjusting resource allocation based on changing transaction volumes. This automated scaling lets the system expand from a baseline of just 4 GPU MIG slices during normal activity to a full deployment of 64 slices during peak processing windows, with the entire scaling operation finishing within two minutes. This responsiveness lets banks maintain rock-solid fraud detection performance during sudden transaction spikes—like those hitting during major

shopping events or holiday periods—without over-provisioning infrastructure for everyday operation. Industry folks have documented that "GPU virtualization techniques, particularly MIG, are crucial for optimizing AI workloads in Kubernetes environments," enabling efficient resource allocation and improved cluster utilization for financial services workloads that see wildly variable demand patterns throughout the business cycle [9].

The throughput and latency characteristics demonstrated by the system highlight the game-changing impact of GPU acceleration on fraud detection workloads. Sustaining 150,000 inferences per second with a 95th percentile latency of just 38 milliseconds represents performance that would demand a massively larger deployment of traditional CPU-based infrastructure, typically 5-10× larger depending on the specific models used. This performance profile lets financial institutions process the complete transaction volume of even the biggest global payment networks without batching or sampling, ensuring every single transaction gets comprehensive fraud analysis without adding noticeable delays to the payment flow. The system's latency distribution looks particularly impressive given the complexity of the models used, with 99.9% of transactions receiving complete fraud assessment—including feature enrichment, multi-stage inference, and, when necessary, graph analysis—within 100 milliseconds. Research shows that Tensor Core technology in modern GPUs offers "both high computational throughput and energy efficiency" for deep learning inference workloads, with mixed-precision operations providing "up to 3x higher performance compared to single-precision" while maintaining the numerical stability required for financial applications [10].

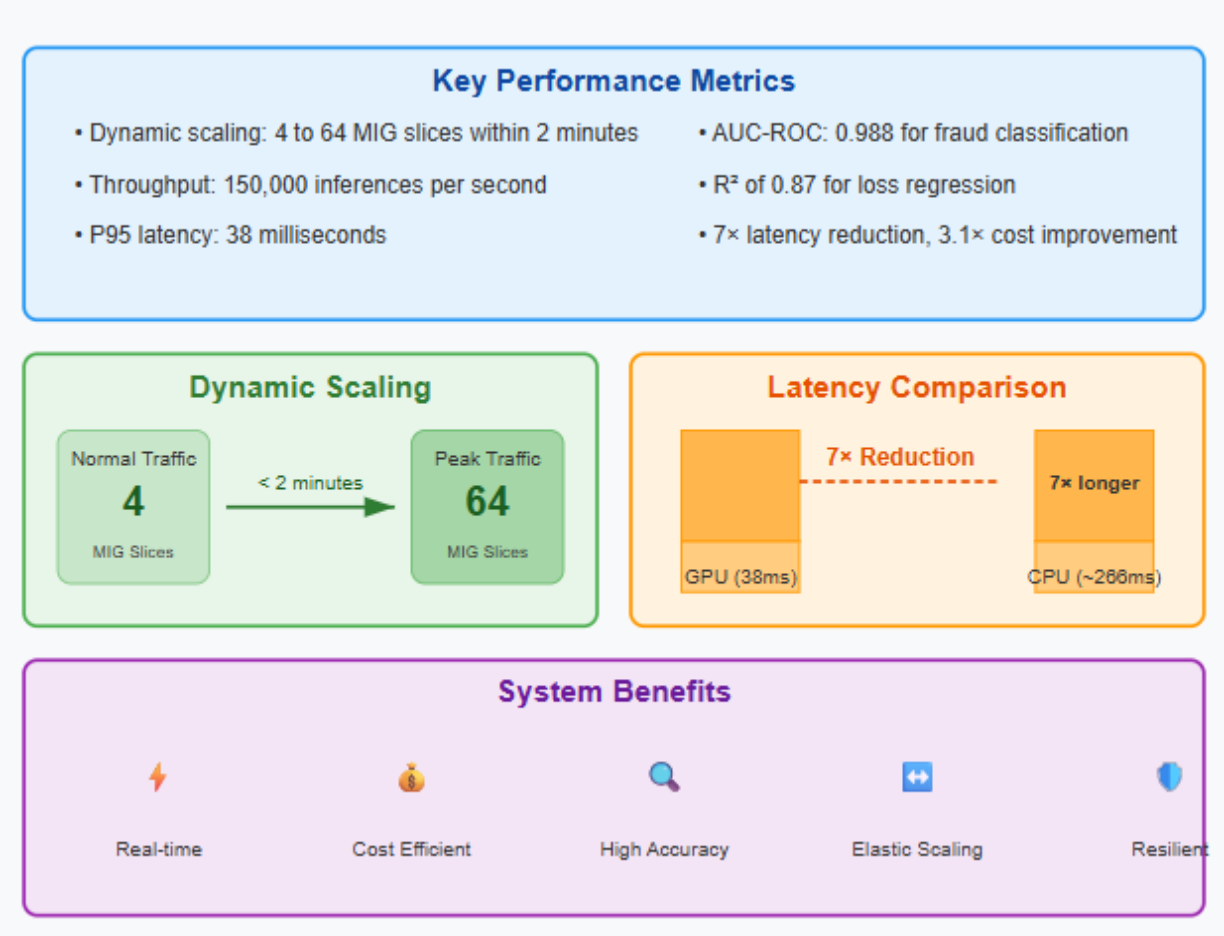

Fig 3: GPU-Accelerated Fraud Detection Performance Metrics [9, 10]

Conclusion

GPU containers give banks serious muscle to fight fraud while staying quick on their feet as criminals constantly switch tactics. Blending cloud-native design with GPU horsepower delivers split-second responses and statistical accuracy crucial for modern financial security without breaking compliance rules or operational flexibility. The cascade approach throws computing power exactly where needed - deep analysis only for sketchy transactions, while rushing legitimate ones through. With auto-scaling, multi-zone deployment, and hardware isolation between workloads, the system delivers knockout performance while staying

rock-solid reliable. Banks don't just get tech wins - they see real business benefits through better fraud catches, fewer false alarms, and much smaller infrastructure bills. As banking goes increasingly digital, this blueprint helps organizations modernize security while handling exploding transaction volumes and ever-smarter fraud schemes. The design shows how smart architecture balances seemingly contradictory goals - speed vs. accuracy, nimbleness vs. stability, innovation vs. compliance - building systems that handle today's needs while adapting to tomorrow's challenges.

Funding: This research received no external funding.
Conflicts of Interest: The authors declare no conflict of interest.
Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] Juhohn Lee, "Credit card fraud will increase due to the Covid pandemic, experts warn," CNBC, 2021. [Online]. Available: https://www.cnbc.com/2021/01/27/credit-card-fraud-is-on-the-rise-due-to-covid-pandemic.html
[2] Naim, Brad Rees, and Summer Liu, "Supercharging Fraud Detection in Financial Services with Graph Neural Networks (Updated)," NVIDIA Corporation, Technical Report, 2025. [Online]. Available: https://developer.nvidia.com/blog/supercharging-fraud-detection-in-financial-services-with-graph-neural-networks/
[3] Nathan Taber and Scott Malkie, "Running GPU-Accelerated Kubernetes Workloads on P3 and P2 EC2 Instances with Amazon EKS," AWS Compute Blog, 2018. [Online]. Available: https://aws.amazon.com/blogs/compute/running-gpu-accelerated-kubernetes-workloads-on-p3-and-p2-ec2-instances-with-amazon-eks/
[4] NVIDIA Corporation, "NVIDIA Multi-Instance GPU," NVIDIA Technologies. [Online]. Available: https://www.nvidia.com/en-in/technologies/multi-instance-gpu/
[5] Jay Kreps et al., "Kafka: a Distributed Messaging System for Log Processing," NetDB Workshop, 2011. [Online]. Available: https://notes.stephenholiday.com/Kafka.pdf
[6] Dinesh Shankar, "Apache Iceberg Explained: A Modern Open Table Format," Medium, 2025. [Online]. Available: https://dishanka.medium.com/apache-iceberg-explained-a-modern-open-table-format-f178334c36ec
[7] Feng Niu et al., "HOGWILD: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent,". [Online]. Available: https://proceedings.neurips.cc/paper/2011/file/218a0aefd1d1a4be65601cc6ddc1520e-Paper.pdf
[8] Tianqi Chen and Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System," KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Pages 785 - 794, 2016. [Online]. Available: https://dl.acm.org/doi/10.1145/2939672.2939785
[9] ServerWala InfraNet FZ-LLC, "AI Workload Optimization Using Kubernetes and GPU Virtualization," Medium, 2025. [Online]. Available: https://medium.com/@serverwalainfra/ai-workload-optimization-using-kubernetes-and-gpu-virtualization-8963d83c2f99
[10] Stefano Markidis et al., "NVIDIA Tensor Core Programmability, Performance & Precision," arXiv:1803.04014, 2018. [Online]. Available: https://arxiv.org/abs/1803.04014