International Journal of Engineering, Science and Information Technology
Volume 6, No. 1 (2026) pp. 146-151
eISSN: 2775-2674
Website: http://ijesty.org/index.php/ijesty
DOI: https://doi.org/10.52088/ijesty.v6i1.1833
Research Paper, Short Communication, Review, Technical Paper

Transforming Real-Time Transactions: Distributed AI, Fraud Detection, and Advanced Analytics in Regulated Financial Environments

Dasaradhi Eddula*

Independent Researcher, United State

**Corresponding author Email: edasaradhi@gmail.com*

The manuscript was received on 2 June 2025, revised on 10 September 2025, and accepted on 4 January 2026, date of publication 24 January 2026

Abstract

The convergence of distributed artificial intelligence (AI), streaming analytics, and real-time payment infrastructure is transforming the financial services sector by enabling faster, more intelligent, and scalable transaction processing. However, this transformation also introduces significant technical, regulatory, and sustainability challenges for financial institutions operating in highly regulated environments. With global fraud losses exceeding USD 485 billion in 2023 and increasing regulatory demands for transparency, fairness, and algorithmic explainability, financial systems must simultaneously achieve high accuracy, low latency, operational efficiency, auditability, and environmental sustainability. This article examines the architectural and operational foundations of compliant distributed AI deployment in modern financial ecosystems. Drawing on current academic and industry literature, the study analyzes the integration of microservices architectures, event-driven processing pipelines, container-based inference engines, edge-cloud computing environments, and federated learning frameworks in supporting real-time fraud detection and payment intelligence. Particular attention is given to fraud detection model architectures, streaming data analytics, distributed model orchestration, and the trade-offs between performance, scalability, and energy consumption. The analysis also explores the broader social and governance dimensions of AI deployment, including fairness, accountability, privacy preservation, regulatory compliance, and carbon-aware computing practices. The findings indicate that well-designed distributed AI systems can achieve fraud-scoring latencies below 20 milliseconds while maintaining high predictive accuracy and compliance with regulatory requirements. Furthermore, federated learning and event-driven architectures enable institutions to improve fraud detection capabilities without centralizing sensitive customer data, thereby enhancing privacy and security. The study concludes that the successful integration of distributed AI and real-time payment infrastructure requires a balanced approach that aligns technological innovation with governance, sustainability, and ethical considerations. The proposed framework offers practical guidance for financial institutions seeking to modernize payment systems while ensuring resilience, trustworthiness, and long-term operational sustainability.

Keywords: *Distributed Artificial Intelligence, Fraud Detection, Machine Learning, Streaming Analytics, Federated Learning*.

1. Introduction

Financial fraud represents one of the most rapidly evolving adversarial domains in modern commerce. Fraud-related losses worldwide hit a record level of $485 billion in 2023, and losses are forecasted to climb above $600 billion per year by 2027, fueled by the rise of online payment channels, the capabilities of criminal gangs, and generative AI technologies that facilitate synthetic identity attacks and large-scale social engineering [1]. Existing systems based on static velocity analysis, threshold policies, and human-curated blocklists cannot adapt quickly enough to dynamically evolving fraud tactics that vary substantially by region, merchant type, and demographic category [2].

Distributed AI systems, trained on hundreds of millions of transactions and deployed at inference latencies below 20 milliseconds, have become the only architecturally viable response. Yet requirements that extend far beyond raw detection accuracy constrain the deployment of AI in regulated financial environments [3]. The European Union's Artificial Intelligence (AI) Act, effective from 2025, classifies credit scoring and fraud detection systems as high-risk AI applications subject to mandatory transparency, bias testing, and human oversight requirements. The United States Consumer Financial Protection Bureau (CFPB) has similarly signaled expectations for algorithmic explainability in adverse action notices [4]. These regulatory imperatives create a tension with the opacity of high-accuracy deep learning models, driving adoption of architectures that balance predictive performance with interpretability. This trade-off has become a central challenge for industry research and engineering [5][6].

Copyright © Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Sustainability adds a third dimension to this challenge. Machine learning inference for fraud detection at the scale of a major payment network consumes substantial compute resources while processing 50,000–100,000 transactions per second during peak periods [7]. Extrapolation from cloud infrastructure benchmarks suggests that a major payment network running continuous neural network inference could consume several hundred megawatt-hours daily [8][9]. Architectural choices that reduce inference energy without degrading detection quality are therefore economically and environmentally material, particularly as regulators begin to mandate Scope 2 emissions reporting for financial institutions' digital operations.

The following section outlines the organization of this paper. Section 2 discusses architecture design philosophies for decentralized AI in finance [10]. Section 3 delves into infrastructure considerations related to efficient inference of deep models. Section 4 covers performance and sustainability engineering in the context of fraud detection and analytics pipelines [11]. Section 5 addresses societal impacts, and Section 6 concludes the paper.

2. Literature Review

2.1. Microservices and Resource Efficiency for Distributed AI

The deployment of distributed AI in real-time financial transaction systems requires architectural decomposition that separates the concerns of feature engineering, model inference, decision orchestration, and human review into independently scalable services [12]. This microservices approach is not merely an engineering preference for AI workloads; it is an operational necessity. Feature engineering pipelines, which compute velocity features, network graphs, and behavioral embeddings from raw transaction streams, have fundamentally different scaling characteristics from model inference services. As transaction volumes are high, the inference workload needs to be scaled quickly enough to preserve Service Level Agreement (SLA) latency. In contrast, feature pipelines require gradual scaling due to the window-aggregation requirements. Blurring the lines between those two issues and building a monolithic AI system results in scaling inefficiency and excessive energy use [13].

The heterogeneity of underlying hardware further amplifies the sustainability benefits of microservice decomposition for AI workloads. Model inference for deep neural networks benefits enormously from Graphics Processing Unit (GPU) acceleration, while CPU-optimized instances execute feature engineering and rule evaluation more efficiently [14]. A microservices architecture allows institutions to co-locate inference services with GPU resources and feature pipelines with CPU-dense nodes, matching hardware to workload profile with a precision that is architecturally impossible in monolithic AI systems. Production deployments of this pattern report GPU utilization rates of 65–80% in well-tuned inference services, compared to 20–35% in monolithic AI deployments where GPU resources are shared across heterogeneous workloads [15].

The modularity of microservices also enables continuous model improvement without disrupting the system. In fraud detection, where adversarial patterns evolve on timescales of days or weeks, the ability to deploy updated model versions to a single inference microservice—using canary or blue-green deployment strategies—without redeploying the entire AI stack is operationally critical [16]. Rolling model updates reduce the blast radius of model degradation incidents, limit the duration of elevated false-positive rates that damage customer experience, and reduce the energy consumed by full-system redeployments. Across a major payment network, the cumulative energy savings from incremental model deployment versus full-system restart cycles can amount to thousands of megawatt-hours annually.

Table 1. Fraud Detection Model Architectures: Performance vs. Interpretability Trade-offs

Model Type	Detection AUC	Inference Latency	Explainability	Energy per Inference
Rule-Based System	0.72–0.80	<1 ms	Full (human-readable rules)	Negligible
Gradient Boosted Trees (XGBoost/LightGBM)	0.88–0.93	1–5 ms (CPU)	High (SHAP values)	Low
Shallow Neural Network	0.91–0.95	2–8 ms (CPU)	Moderate (LIME)	Low–Medium
Deep Neural Network (MLP/Transformer)	0.95–0.98	8–25 ms (GPU)	Low (post-hoc XAI required)	High
Ensemble (GBT + DNN)	0.96–0.99	10–30 ms (GPU)	Moderate (component-level)	High
Federated GBT	0.86–0.91	2–6 ms (CPU)	High (SHAP on local model)	Low (distributed)

2.2. Event-Driven Systems for Real-Time Fraud Analytics

Event-driven architectures are the natural substrate for real-time fraud detection systems, because fraud signals are fundamentally event-driven [17]. Each transaction event carries features that, when combined with historical context, either corroborate or challenge its legitimacy. The key architectural challenge is enriching each transaction event with the contextual features needed for accurate scoring within the sub-20-millisecond latency budget required by authorization networks. Event-driven architectures address this challenge through stateful stream processing: as each transaction event flows through the streaming platform, it is enriched with precomputed features retrieved from distributed feature stores, scored by inference services that consume from the event stream, and forwarded to downstream decision and notification services based on the score result [18].

The energy conservation properties of event-driven fraud detection systems are substantial. Polling-based fraud systems consume CPU and network resources continuously, regardless of transaction arrival rates, by periodically querying a transaction database for new records to score. In off-peak periods when transaction volumes may drop to 5–10% of peak, polling-based systems continue to consume near-peak resources while accomplishing near-zero useful work. Event-driven systems, by contrast, scale their compute consumption proportionally to transaction volume through consumer group autoscaling, achieving energy reductions of 40–65% during low-volume periods without any degradation in detection capability during peak periods [19].

Federated learning represents an important extension of event-driven AI architectures for regulated financial environments. Rather than centralizing transaction data from multiple participating institutions—which raises concerns about data sovereignty, privacy, and

competitive sensitivity—federated learning enables distributed model training across institution boundaries, exchanging only model gradients rather than raw data. Research on federated learning for fraud detection has demonstrated that federated models trained across multiple institutions can achieve detection accuracy within 2–5% of centralized models trained on pooled data, while eliminating the data-sharing requirements that regulatory frameworks such as the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA) would otherwise complicate [20]. At the persistence layer, autonomous multi-zone replication architectures extend this resilience model by maintaining zero-loss commit guarantees across distributed cloud zones, ensuring that AI-processed fraud decisions and compliance events remain intact even during complex multi-zone infrastructure failures [21].

3. Methods

3.1. Containerized Model Serving and Resource Density

Containerized AI model serving is a major improvement in the efficiency of fraud detection infrastructure operation. Before containerization, inference in financial organizations was typically handled by server clusters running proprietary inference engines with preconfigured resources. Such a process was highly inefficient because servers assigned to specific model versions were unable to handle different tasks once a new model version appeared; moreover, the resources assigned to those versions were calculated based on peak usage rather than actual demand. By adopting containerization, resource efficiency increased because inference containers could be launched based on the number of tasks in the queue; thus, the ratio of used to available resources reached 65–80%, compared to 20–40% in non-containerized inference services [15].

The process of scaling out inference through containerization introduces additional challenges related to model lifecycle management, dependency management, and environmental stability. Such problems can be easily addressed by adopting container immutability, where each version of a model becomes a separate containerized image that includes all runtime dependencies and model data. Furthermore, the reproducibility guarantee provided by container images meets the requirements for model governance set by guidelines such as SR 11-7 (Supervisory Guidance on Model Risk Management). In such frameworks, production model environments must be fully reproducible and well-documented for regulatory inspection.

GPU resource sharing via containerized inference offers additional opportunities for efficiency gains. Modern container orchestration platforms support fractional GPU allocation, enabling multiple lightweight inference services to share a single physical GPU rather than requiring dedicated GPU instances per model. In fraud detection architectures that run separate models for different transaction categories—card-present, card-not-present, Automated Clearing House (ACH), wire transfer—fractional GPU sharing can reduce GPU instance counts by 40–60% compared to dedicated-instance deployments, with proportional reductions in energy consumption and infrastructure cost [22].

Table 2. AI Infrastructure Optimization Strategies and Performance Impact

Strategy	Technique	Resource Saving	Regulatory Benefit
Containerized Serving	Immutable inference images	GPU utilization 65–80% vs. 20–40%	SR 11-7 reproducibility compliance
Fractional GPU Sharing	Container GPU slicing	40–60% fewer GPU instances	Cost and carbon reduction
Priority-Aware Scheduling	quality of service classes + node affinity	P99 latency reduced 15–30%	Inference SLO guaranteed
Spot Instances (Batch)	Preemptible training nodes	60–80% compute cost reduction	Indirect — lower Scope 2 emissions
Federated Training	Gradient aggregation across sites	Eliminates data centralization	GDPR/CCPA compliance

3.2. Orchestration for AI Workload Lifecycle Management

The orchestration of AI workloads in financial transaction systems poses challenges qualitatively distinct from those of general-purpose microservice orchestration. Model inference services have latency Service Level Objectives (SLOs) measured in the single-digit milliseconds range. In contrast, model training and feature backfill jobs can tolerate much higher latency in exchange for lower cost. Effective orchestration must simultaneously guarantee the latency SLOs of inference services and maximize the energy efficiency of training and batch analytics jobs—a dual objective that requires workload classification and priority-aware scheduling policies that most general-purpose orchestration configurations do not implement by default [23].

Priority-aware scheduling keeps latency-critical inference services separate from batch analytics jobs by using Kubernetes quality-of-service (quality of service) classes and node affinity rules. This ensures that inference pods run on high-performance nodes with guaranteed CPU and memory, while batch jobs run on preemptible spot instances that are 60–80% cheaper and produce less carbon than on-demand compute. This separation is particularly important during model training cycles, which can consume 10–50x the compute of steady-state inference and would severely degrade authorization latency if allowed to compete for resources on shared infrastructure. Institutions implementing priority-aware AI workload orchestration report reductions in authorization tail latency of 15–30% at P99 compared to shared-resource configurations [24].

Continuous model monitoring and automated retraining pipelines represent advanced orchestration capabilities that are increasingly essential in adversarial fraud environments. As fraud patterns shift, model performance degrades—a phenomenon known as concept drift—and detection rates decline without triggering any infrastructure alert. Automated monitoring of model performance metrics, including precision, recall, and false-positive rate, with triggered retraining when metrics cross predefined thresholds, enables institutions to respond to concept drift within hours rather than days or weeks. The orchestration of these retraining pipelines benefits substantially from dedicated workflow engines such as Kubeflow Pipelines or MLflow, which provide the dependency management, reproducibility, and audit logging required in regulated environments. Alongside orchestration, predictive failure modeling that relies on telemetry turns database

infrastructure data—such as transaction latencies, buffer cache usage statistics, and synchronization data—into actionable signals that detect potential issues before they affect pipeline performance or data availability for subsequent fraud scoring [25].

4. Result and Discussion

4.1. Caching for Feature and Model Output Reuse

Two operations—feature computation and model inference—dominate the computational cost of real-time fraud detection. Both are amenable to caching strategies that dramatically reduce per-transaction energy consumption without materially degrading detection quality. Feature caching stores frequently accessed behavioral features—customer transaction velocity, device fingerprints, merchant risk profiles, and network graph embeddings—in distributed in-memory stores such as Redis, making them available for retrieval in under one millisecond rather than recomputing them from raw event history on each transaction. In production fraud detection systems processing 50,000 transactions per second, feature caching reduces database load by 60–75% and cuts feature computation latency from 15–40 milliseconds to under 2 milliseconds for cached features [26].

Model output caching—storing fraud scores for frequently recurring transaction profiles—provides an additional layer of energy efficiency for the subset of transactions amenable to this treatment. Low-risk routine transactions, such as a known customer's daily coffee purchase at a familiar merchant using a previously verified device, exhibit negligibly different fraud risk profiles across occurrences. Caching the fraud score for such profiles, with an adaptive Time-To-Live (TTL) that shortens when account activity changes and lengthens during periods of behavioral stability, can reduce live model inference volume by 20–35% without a meaningful impact on fraud detection rates [27]. The energy implications of this reduction are material: a 25% reduction in inference volume across a network processing 65,000 peak transactions per second translates to approximately 16,000 fewer GPU inference operations per second—a significant reduction in energy consumption at scale.

The design of caching systems for fraud detection requires careful attention to the adversarial dimension of the problem domain. Fraudsters who understand cache architectures can attempt to exploit cache TTLs by spacing fraudulent transactions within a window where the cached "safe" score from a prior transaction remains valid. Effective cache design in fraud systems, therefore, combines behavioral TTL policies—which expire cache entries when account behavior deviates from the profile used to generate the cached score—with rate-of-change monitoring that detects anomalous transaction patterns that might indicate cache exploitation. This adversarial-aware cache design adds modest implementation complexity while preserving the energy-efficiency benefits of caching without creating exploitable detection blind spots.

Table 3. Model Compression Techniques for Fraud Detection: Accuracy vs. Efficiency

Technique	Accuracy Impact	Inference Speedup	Memory Reduction	Best Use Case
INT8 Quantization	<0.5% AUC loss	2–3x	4x	Edge/mobile fraud scoring
INT4 Quantization	0.5–2% AUC loss	3–4x	8x	High-volume batch scoring
Weight Pruning	<1% AUC loss (unstructured)	1.5–2x	50–70%	Sparse architecture serving
Knowledge Distillation	<2–5% AUC loss	5–10x	80–90%	Latency-critical real-time scoring
Tiered Inference	Negligible at the system level	40–60% avg cost reduction	N/A	End-to-end production pipelines

4.2. Computation Efficiency and Model Compression

The compute resources consumed per inference operation directly determine the energy efficiency of AI fraud detection. Research on anomaly detection and fraud scoring models has demonstrated that substantial accuracy preservation is achievable through model compression techniques that reduce inference cost by 40–70% [28]. Quantization—representing model weights and activations in 8-bit or 4-bit integers rather than 32-bit floating point—reduces model memory footprint by 4–8x and inference latency by 2–3x on compatible hardware, with accuracy losses of typically less than 0.5% on financial fraud benchmarks. The technique of knowledge distillation trains a smaller "student" model to mimic the output probability distribution of a larger "teacher" model, resulting in efficient models whose prediction accuracy accounts for 90–95% of the teacher's but costs 20–30% of the inference resources.

The architecture of tiered inference uses model compression at the system level and routes transactions based on their risk profiles. Low-risk transactions, which account for around 70–80% of the traffic, are classified using an efficient gradient-boosted decision tree model. Only 20–30% of high-risk transactions are forwarded to more expensive deep neural networks for processing. This tiered approach achieves an effective average inference cost reduction of 40–60% compared to uniformly applying the heavy model, while maintaining the deep model's detection accuracy for transactions where it matters most [29].

Since 2020, hardware acceleration for inference has advanced significantly, and its application in finance fraud detection is becoming increasingly common. Modern inference accelerators—including NVIDIA A100/H100 GPUs, Google TPU v4, and AWS Inferentia2—deliver 10–50x improvements in throughput per watt compared to CPU-only inference for transformer and Multilayer Perceptron (MLP) architectures commonly used in fraud scoring. The business case for inference accelerators in financial transaction systems is compelling even at current hardware costs: a single H100 GPU can replace 40–80 CPU cores to achieve equivalent inference throughput with deep fraud detection models, reducing both capital expenditure and per-inference energy consumption by comparable amounts.

Table 4. Streaming Analytics Pipeline Benchmarks for Real-Time Fraud Detection

Pipeline Component	Technology	Throughput	Latency (P99)	Energy Characteristic
Event Ingestion	Apache Kafka	>1M events/sec/cluster	<5 ms (broker)	Low — event-driven activation
Feature Computation	Apache Flink (stateful)	200K–500K TPS	5–15 ms	Medium — stateful aggregation
Feature Retrieval (cache)	Redis Cluster	>1M ops/sec	<1 ms	Very low — memory lookup
Model Inference (GPU)	TensorRT / ONNX Runtime	50K–200K inf/sec per GPU	3–8 ms	High — GPU-accelerated
Decision Orchestration	Custom microservice	>100K decisions/sec	<2 ms	Low — rule evaluation
Audit Logging	Apache Kafka + HDFS	>500K records/sec	Async — no latency impact	Low — sequential write

4.3. Broader Societal Implications

While discussing the impact of incorporating distributed AI algorithms into financial transaction regulation systems, it is necessary to address the broader societal issues they raise. AI-based fraud detection systems make important decisions about whether to allow or block particular transactions, which significantly affect customers' ability to use their funds and to perform e-commerce. When biased AI systems lead to a decline in customer transactions in certain geographic locations, age cohorts, or income levels, this translates into real-life discrimination, resulting in financial losses and damage to brand reputation [30]. Machine learning fairness research indicates that models trained on historical fraud data might amplify existing inequalities in fraud investigations and create a bias feedback loop that favors those who have been investigated more often.

Rapidly changing regulatory guidelines reflect the concerns mentioned above. The EU AI Act requires bias testing for its high-risk fraud detection category. The CFPB's adverse action requirements require that declined transactions be accompanied by explanations meaningful to consumers—a requirement that deep learning models cannot satisfy without post-hoc explainability tools such as Shapley Additive exPlanations (SHAP) or Local Interpretable Model-Agnostic Explanations (LIME). The architectural implication is clear: production fraud detection systems must incorporate explainability as a first-class system capability, not an afterthought appended after model development [31].

Environmental responsibility in AI-powered financial systems is increasingly a regulated obligation rather than a voluntary commitment. Data centers supporting AI inference workloads for major payment networks consume substantial electricity, and the financial services sector faces growing pressure from both regulators and institutional investors to quantify and reduce the carbon intensity of digital operations. The architectural and engineering strategies described in this article—tiered inference, model compression, federated learning, and energy-aware orchestration—represent a technically mature toolkit for constraining the growth of AI energy consumption in financial systems. Their adoption at scale would meaningfully reduce the carbon footprint of an industry that is simultaneously expanding its AI capabilities and facing regulatory pressure to demonstrate environmental stewardship [32].

5. Conclusion

Distributed AI has become indispensable for detecting, preventing, and investigating financial fraud at the scale and speed required by modern payment networks. This article has demonstrated that the architectural, infrastructure, and performance engineering choices that determine AI effectiveness in financial systems also determine energy efficiency. Microservice decomposition enables hardware-workload matching, improving GPU utilization from 20–40% to 65–80%. Event-driven pipelines eliminate polling-based energy waste and provide the immutable audit trails that regulatory compliance requires. Containerized, priority-aware orchestration maintains sub-20-millisecond inference latency while enabling 60–80% cost reductions for batch training workloads through intelligent spot instance utilization.

Model compression techniques—quantization, distillation, and pruning—and tiered inference architectures collectively reduce per-inference energy consumption by 40–70% with minimal impact on accuracy, resolving the apparent tension between detection quality and sustainability. These technical achievements, however, do not resolve the deeper societal questions raised by the embedding of AI in financial decision-making. Fairness, explainability, and accountability require architectural commitment—not as retrofitted compliance features but as foundational design principles that shape model selection, training data governance, and deployment monitoring from inception.

As the regulatory landscape for AI in financial services continues to evolve toward greater specificity and enforceability, institutions that have built explainability and bias testing into their AI infrastructure architectures will be positioned to adapt with lower compliance costs and higher public trust than those that have treated these dimensions as secondary concerns. Future research directions may include developing methods to measure the energy usage characteristics of financial AI inference tasks, studying federated learning performance over time in real-world multi-institutional fraud scenarios, and evaluating fairness concerns with model compression approaches.

References

[1] K. RV, S. Ganesh, B. GR, and M. Thiruvengadam, "Credit card fraud data analysis and prediction using machine learning algorithms," *Secur. Priv.*, vol. 8, no. 3, p. e70043, 2025.

[2] M. Hole\vna, P. Pulc, and M. Kopp, *Classification methods for internet applications*. Springer, 2020.

[3] S. K. Aljunaid, S. J. Almheiri, H. Dawood, and M. A. Khan, "Secure and transparent banking: Explainable AI-driven federated learning model for financial fraud detection," *J. Risk Financ. Manag.*, vol. 18, no. 4, p. 179, 2025.

[4] J. B. Rajendra and A. S. Thuraisingam, "The role of explainability and human intervention in AI decisions: jurisdictional and regulatory aspects," *Inf. \& Commun. Technol. Law*, vol. 35, no. 2, pp. 152–183, 2026.
[5] P. Selvakumar, C. Singh, S. K. Basu, T. C. Manjunath, S. N. Shalini, and R. K. Mishra, "Challenges of Explainable AI in Security Decision-Making," in *Intelligent Cyber Defense for Critical Infrastructure*, IGI Global Scientific Publishing, 2026, pp. 87–114.
[6] F. Hariyanto, T. Budiman, A. B. Yulianto, and V. Yasin, "Designing a web-based information system for monitoring final projects," *Int. J. Eng. Sci. Inf. Technol.*, vol. 5, no. 2, pp. 142–153, 2025.
[7] K. Yu, Y. Chen, T. K. Trinh, and W. Bi, "Real-time detection of anomalous trading patterns in financial markets using generative adversarial networks," 2025.
[8] Y. Zhu, "Study on the Full Life-Cycle Environmental Impact and Systematic Optimization of Large Models," *Front. Econ. Manag.*, vol. 7, no. 3, pp. 176–188, 2026.
[9] A. K. A. JANNAH, N. A. R. EFRAT, and R. Permatasari, "Ui/Ux Design Of Ruang Kajian Application Using Design Thinking Method," *Int. J. Eng. Sci.*, vol. 5, no. 1, pp. 145–156, 2024.
[10] M. S. Al Jasem, T. De Clark, and A. K. Shrestha, "Toward decentralized intelligence: A systematic literature review of blockchain-enabled AI systems," *Information*, vol. 16, no. 9, p. 765, 2025.
[11] S.-A. Ionescu, V. Diaconita, and A.-O. Radu, "Engineering sustainable data architectures for modern financial institutions," *Electronics*, vol. 14, no. 8, p. 1650, 2025.
[12] R. Joshi, "Human-in-the-loop AI in financial services: Data engineering that enables judgment at scale," *J. Comput. Sci. Technol. Stud.*, vol. 7, no. 7, pp. 228–236, 2025.
[13] J. Jiang, Q. Li, P. Wang, and Y. Liu, "DRKC: Deep Reinforcement Learning Enhanced Microservice Scheduling on Kubernetes Clusters in Cloud-Edge Environment," *IEEE Trans. Cloud Comput.*, 2025.
[14] D. Danopoulos, "Hardware-software co-design of deep learning accelerators: from custom to automated design methodologies," 2024.
[15] L. Zhu, K. Huang, Y. Hu, and Y. Wang, "Two-Stage Learning Approach for Semantic-Aware Task Scheduling in Container-Based Clouds," *IEEE Trans. Cloud Comput.*, vol. 13, no. 1, pp. 148–165, 2024.
[16] Z. Zamanzadeh Darban, G. I. Webb, S. Pan, C. Aggarwal, and M. Salehi, "Deep learning for time series anomaly detection: A survey," *ACM Comput. Surv.*, vol. 57, no. 1, pp. 1–42, 2024.
[17] K. R. Gade, "Event-driven data modeling in fintech: A real-time approach," *J. Comput. Innov.*, vol. 3, no. 1, 2023.
[18] C. Shan, Y. Xia, Y. Zhan, and J. Zhang, "KubeAdaptor: A docking framework for workflow containerization on Kubernetes," *Futur. Gener. Comput. Syst.*, vol. 148, pp. 584–599, 2023.
[19] Y. Qiao, S. Shen, C. Zhang, W. Wang, T. Qiu, and X. Wang, "EdgeOptimizer: A programmable containerized scheduler of time-critical tasks in Kubernetes-based edge-cloud clusters," *Futur. Gener. Comput. Syst.*, vol. 156, pp. 221–230, 2024.
[20] J. M. Parra-Ullauri *et al.*, "kubeFlower: A privacy-preserving framework for Kubernetes-based federated learning in cloud--edge environments," *Futur. Gener. Comput. Syst.*, vol. 157, pp. 558–572, 2024.
[21] F. Z. Gargab, "Technical-economic optimization and manufacture of a Moroccan solar water heater." Université de Pau et des Pays de l'Adour; Université Sidi Mohamed ben~…, 2021.
[22] G. Pang, C. Aggarwal, C. Shen, and N. Sebe, "Editorial deep learning for anomaly detection," *IEEE Trans. Neural Networks Learn—Syst.*, vol. 33, no. 6, pp. 2282–2286, 2022.
[23] J. E. de Albuquerque Filho, L. C. P. Brandao, B. J. T. Fernandes, and A. M. A. Maciel, "A review of neural networks for anomaly detection," *IEEE Access*, vol. 10, pp. 112342–112367, 2022.
[24] X. Wang, L. Duan, L. Guan, J. Xu, and C. He, "Learning Decision Boundaries for Multidimensional Anomaly Detection," *IEEE Trans. Neural Networks Learn. Syst.*, 2025.
[25] R. Gollapudi, "Telemetry-Driven Predictive Failure Models for High-Scale Financial Databases," *J. Comput. Anal. Appl.*, vol. 34, no. 12, 2025.
[26] S. Cai and Z. Xie, "Explainable fraud detection of financial statement data driven by a two-layer knowledge graph," *Expert Syst. Appl.*, vol. 246, p. 123126, 2024.
[27] A. T. Khan, X. Cao, S. Li, V. N. Katsikis, I. Brajevic, and P. S. Stanimirovic, "Fraud detection in publicly traded US firms using Beetle Antennae Search: A machine learning approach," *Expert Syst—Appl.*, vol. 191, p. 116148, 2022.
[28] V. L. Narayana Gorle and S. Panigrahi, "An efficient heuristic optimization-based fraudulent activities detection in the financial sector using adaptive machine learning and deep learning system," 2026.
[29] A. Taghinezhad-Niar and J. Taheri, "Reliability, rental-cost, and energy-aware multi-workflow scheduling on multi-cloud systems," *IEEE Trans. Cloud Comput.*, vol. 11, no. 3, pp. 2681–2692, 2022.
[30] C.-H. Lu, "The moderating role of e-lifestyle on disclosure intention in mobile banking: A privacy calculus perspective," *Electron. Commer. Res. Appl.*, vol. 64, p. 101374, 2024.
[31] M. A. Adnan and others, "A prediction-based replica selection strategy for reducing tail latency in geo-distributed systems," *IEEE Trans. Cloud Comput.*, vol. 11, no. 3, pp. 2954–2965, 2023.
[32] J. O. Ojadi, E. Onukwulu, C. Odionu, and O. Owulade, "Leveraging IoT and deep learning for real-time carbon footprint monitoring and optimization in smart cities and industrial zones," *IRE Journals*, vol. 6, no. 11, pp. 946–964, 2023.