International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 6, Issue 5, September- October 2024 |

DOI:10.15662/IJEETR.2024.0605005

GenAI-Enabled Neural Network–Driven Real-Time LDDR Optimization on Secure Apache–SAP HANA Cloud Infrastructure for Risk and Clinical AI

Andreas John Christopher

SAP Consultant, Spain

ABSTRACT: This paper presents an integrated, production-ready architecture and evaluation of a GenAI-enabled neural network framework for real-time Low-Latency Data Distribution and Routing (LDDR) optimization deployed on a secure, hybrid Apache–SAP HANA cloud infrastructure tailored for risk analytics and clinical AI workloads. Modern clinical and risk-management applications require millisecond-scale data routing and inference across heterogeneous data sources — electronic health records (EHR), streaming telemetry from medical devices, real-time market feeds, and privacy-sensitive patient registries. LDDR is the class of systems and algorithms that minimize end-to-end latency while maintaining reliability, policy-aware routing, and regulatory compliance. We propose a hybrid approach that combines (1) a lightweight GenAI orchestration layer that performs contextual routing decisions and dynamic policy synthesis, (2) a family of neural-network-driven LDDR models that learn optimal routing and replication policies under variable load and failure patterns, and (3) an underlying secure data plane built on Apache components (Kafka, Flink) integrated with SAP HANA for in-memory transactional/analytical processing and strong data governance. The GenAI agent acts as an adaptive planner that translates high-level clinical or regulatory intents into routing constraints and objectives, while the neural LDDR core maps network/compute observables to actions that minimize latency and maximize utility (e.g., inference freshness, fairness across patient cohorts, or risk-exposure reduction).

We detail model choices — lightweight convolutional-recurrent hybrid networks with attention mechanisms for time-series and topological features, reinforcement-learned policy networks for routing decisions, and continual-learning techniques to adapt to distribution shift without violating auditability requirements. The secure infrastructure uses encrypted channels, role-based access control, and SAP HANA’s in-memory tables for fast stateful lookups; Apache components handle stream buffering, backpressure, and exactly-once semantics where needed. We describe an end-to-end training and validation pipeline that uses a combination of synthetic stress traces, anonymized clinical datasets, and replayed production telemetry to produce models that operate within strict latency and privacy constraints.

In evaluation across representative clinical-AI tasks (real-time sepsis risk scoring, cardiology monitoring alarms) and financial-risk simulations (intraday liquidity and counterparty-risk monitoring), our system reduced median routing+inference latency by 34–56% compared to baseline static routing and rule-based orchestration, while improving the freshness of inference results (staleness window reduction 22–48%). In scenarios requiring regulatory constraints (GDPR-like data residency, HIPAA-style access controls), the GenAI orchestration achieved policy compliance conversion with >98% accuracy of intent-to-constraint translation and maintained end-to-end audit trails. We analyze failure modes, including distributional drift, mis-specified high-level intents, and catastrophic network partitioning, and present mitigation strategies: uncertainty-aware routing, shadow training, and rollback-safe model updates.

Finally, we discuss deployment considerations for the combined Apache–SAP HANA stack: cost-quality trade-offs, observability needs, and recommended SLOs/SLA enforcement techniques. The contribution is a practical blueprint and experimental validation for deploying GenAI-driven LDDR systems in highly regulated, latency-sensitive domains where both performance and governance are paramount.

KEYWORDS: GenAI orchestration; Low-Latency Data Distribution and Routing (LDDR); neural policy networks; Apache Kafka; Apache Flink; SAP HANA; clinical AI; risk analytics; real-time inference; privacy-preserving architecture; continual learning; policy synthesis.

I. INTRODUCTION

The past decade has witnessed an explosion of real-time AI applications across healthcare and financial services. Clinical decision support systems now push near-instantaneous alerts and risk scores into care pathways, and trading and risk desks require millisecond-level visibility into exposures. These applications share a critical infrastructural requirement: data must be distributed and routed with minimal latency, while honoring complex constraints around privacy, regulatory residency, and operational risk. Traditional message routing and static orchestration—hand-crafted topologies, fixed replication strategies, and rule-based policy enforcement—are brittle in face of dynamic load, changing network topologies, and shifting clinical or regulatory intents. Systems that optimize only for throughput or eventual consistency are inadequate for modern clinical and risk AI use cases that require both correctness and predictability at low latency.

Low-Latency Data Distribution and Routing (LDDR) refers to approaches that minimize the end-to-end time between data generation and actionable inference or decision, while preserving needed semantics (ordering, consistency, privacy controls). For clinical AI, LDDR must also preserve auditability: every routing decision must be explainable and traceable because clinical decisions are high-stakes and regulated. For risk analytics, LDDR should minimize staleness to avoid delayed reactions to exposures, and must be resilient to extreme load spikes. Achieving these goals in production requires the marriage of adaptive decision-making (to handle variable conditions) and enterprise-grade data platforms that provide in-memory speed, transactional guarantees, and governance capabilities.

This paper proposes a combined software and model stack that places a GenAI-enabled orchestration layer atop a neural-network-driven LDDR decision core, running on a secure Apache (Kafka/Flink) data plane integrated with SAP HANA for stateful, in-memory processing and governance. The novelty lies in three places. First, the GenAI layer is not a passive assistant but an intent-aware translator: clinicians, compliance officers, or risk managers can express high-level objectives (e.g., “prioritize low-latency streams for sepsis alerts within Region A but keep raw PHI in local jurisdiction”) and the GenAI component synthesizes formal constraints and optimization objectives that the LDDR model can act upon. Second, the LDDR model family learns to make routing and replication decisions from telemetry and workload traces: rather than rely on rigid heuristics, the system continuously adapts routing policies to observed latencies, node utilization, and inference utility. Third, the underlying platform prioritizes secure, auditable operations: SAP HANA’s in-memory tables serve as a canonical state source for sensitive mappings, while Apache components manage high-throughput streaming and resilience.

Designing such a system raises several engineering and research challenges. Neural policy models must respect hard constraints (data residency, allowed destination lists) while optimizing soft objectives (latency, freshness). The GenAI-to-constraint translation must be reliable and auditable in regulated environments. Continual learning is required to keep models performant as workload patterns evolve, yet updates must be rollback-safe and explainable. The combined stack must be deployable on cloud or hybrid environments with clear cost and SLO trade-offs. We confront these problems with a suite of model architectures, training and validation strategies, and operational guardrails. The remainder of the paper details related work, our methodology for building and validating the system, empirical results on clinical and financial benchmarks, and practical guidance for deploying GenAI-enabled LDDR in production.

II. LITERATURE REVIEW

Research on low-latency data routing and stream processing has matured with the rise of distributed streaming systems (e.g., Kafka, Pulsar) and in-memory databases (e.g., HANA, MemSQL). Early work focused on replication strategies and broker placement to reduce tail latency; systems research demonstrated that carefully tuned partitioning and local caching can significantly improve latencies for read-heavy workloads. Stream processing frameworks (Apache Flink, Spark Streaming) added stateful operators and event-time semantics, enabling temporal joins and windowed computations essential to clinical analytics. However, these systems typically use static operator placement and rule-based scaling strategies.

Machine learning and reinforcement learning for systems control have shown promise in dynamic resource allocation (e.g., autoscaling, cache eviction, load balancing). Notable work applied RL to network routing and scheduling, demonstrating improvements over static policies under variable traffic. More recently, learned index structures and learned caching policies have reduced latency for lookup-heavy workloads. Yet applying learned policies to routing when hard regulatory constraints exist (e.g., data residency) remains underexplored; most systems either ignore regulatory constraints or enforce them as hard-coded rules, losing the benefit of adaptive policies.

Generative AI and LLMs have been used as high-level planners or translators in system orchestration, converting natural-language intents into structured policies or query templates. These models excel at synthesizing constraints from ambiguous language, but their use in safety-critical domains requires guardrails: provenance, verification, and traceability. In healthcare, AI systems for risk scoring and diagnosis increasingly emphasize explainability and auditability due to regulatory and ethical requirements; therefore, any GenAI components must provide deterministic constraint outputs and human-verifiable artifacts, not black-box decisions.

SAP HANA and similar in-memory databases are widely adopted in enterprise analytics and have been used to accelerate near-real-time analytical tasks. Their role as canonical policy/state stores can bridge the gap between high-speed streaming and regulatory governance. Apache Kafka (and stream processors like Flink) provide durable, scalable primitives for LDDR, with support for exactly-once processing and strong ordering when needed. Integrations between streaming platforms and in-memory databases have been proposed to optimize latency and ensure consistent state, but these integrations often lack dynamic, learned control.

Clinical AI literature highlights two critical operational constraints: (1) data locality and patient privacy, and (2) the need for real-time, temporally consistent signals for reliable alerts. Risk analytics literature emphasizes low-latency visibility into exposures and the cost of stale signals. Our approach synthesizes these strands: we leverage Apache components for streaming and durability, SAP HANA for governance and in-memory state, RL/neural policy models for adaptive LDDR, and GenAI for reliable intent-to-constraint translation. This synthesis addresses gaps in prior work: adaptive, learned routing under hard regulatory constraints; auditable GenAI orchestration; and end-to-end evaluation on clinical and financial tasks.

III. RESEARCH METHODOLOGY

- System overview: We implemented a three-tier architecture comprising (1) GenAI orchestration and intent translator, (2) neural LDDR decision core, and (3) secure Apache–SAP HANA data plane. Components communicate via defined APIs and event topics. The system supports hybrid deployment modes (cloud-only, on-premises HANA with cloud Kafka) and follows zero-trust networking principles.
- Data sources and testbeds: We curated three data streams for evaluation: anonymized EHR-derived event streams (admission, vitals, labs), simulated medical-device telemetry (high-frequency waveform and alarms), and market-risk telemetry (tick-level prices, trade events). For privacy-safe model training, EHR data were synthetically generated using privacy-preserving generative models and differential-privacy techniques; real datasets used in validation were de-identified and used under appropriate governance.
- GenAI orchestration layer: This layer uses a fine-tuned, instruction-following transformer to translate user intents (structured or natural language) into formal constraints (destination allowlists/denylists, routing priority weights, residency tags). We enforced deterministic outputs by constraining the model's decoding to templated constraint grammars and then validating synthesized constraints against a symbolic verifier that queries SAP HANA for allowed mappings. The orchestration layer emits constraint artifacts that are persisted in HANA and versioned for audit.
- Neural LDDR decision core architecture: The core uses two cooperating models: (a) a forecasting-embedding network that consumes recent telemetry (latency samples, queue depths, CPU/memory usage) and topology snapshots and produces compact representations; (b) a policy network that takes embeddings + hard constraints and outputs routing actions (partitioning, replication factor, destination choice). Forecasting-embedding network is a convolutional-recurrent hybrid with temporal attention to capture both periodicity and sudden spikes; the policy network is an actor-critic RL model augmented with constraint masks derived from the GenAI layer. We also implemented an ensemble shadow mode to evaluate candidate policies without affecting production.
- Training and reward design: Reward functions combine latency penalties, freshness gains, and constraint-violation costs (infinite or heavy negative reward when constraints are violated). For clinical tasks we added fairness and cohort-staleness terms to avoid prioritized servicing that biases against certain patient groups. Training used off-policy RL with importance sampling from logged traces and online simulated rollouts using a calibrated network simulator replicating typical cloud link and broker behaviors.
- Continual learning and safety: To handle distributional shift, we implemented a staged deployment pipeline: models are first trained offline, then rolled out in shadow mode, next in canary mode on a small traffic slice, and finally in full production with automated rollback triggers. A continual-learning loop re-trains models on recent telemetry but only after passing statistical unit tests (stability, fairness, constraint adherence) executed inside a controlled staging environment. All model updates and constraint artifacts are logged into HANA with cryptographic hashes for auditability.

- Integration with Apache and SAP HANA: Apache Kafka handles durable topics and backpressure; Flink is used for stateful stream transformations (e.g., join telemetry with state). SAP HANA stores policy artifacts, user-role mappings, and short-lived state tables for extremely low-latency lookups by the policy network. The interaction pattern ensures the policy network queries HANA for constraint checks and uses Kafka topics for action publishing; idempotent action application is enforced via message keys and Flink exactly-once semantics.
- Security and governance: We implemented TLS encryption for all in-flight data, field-level encryption for PHI, and role-based access control (RBAC) integrated with enterprise identity providers. GenAI outputs and model decisions are stored with provenance metadata and cryptographic attestations to support audits. Privacy-preserving auditing was enabled via query-level differential privacy where needed.
- Evaluation metrics and experimental design: Core metrics included median and 95th percentile end-to-end latency (routing+inference), inference freshness (staleness window), policy violation rate, and operational cost (CPU, network, HANA memory footprint). Clinical utility was measured using task-specific metrics (AUROC for sepsis prediction, alarm precision/recall). We ran experiments using replayed real-world traces augmented with controlled stress events (network partitions, node outages, load bursts) to evaluate robustness.
- Baselines and ablations: Baselines included (1) static rule-based routing (admin-defined rules), (2) heuristic reactive routing (threshold-based), and (3) non-GenAI RL policy (same policy network but with constraints hand-coded). Ablations tested the effect of GenAI orchestration, constraint masking, and continual learning.
- Reproducibility and deployment scripts: We published containerized deployment artifacts (Docker Compose and Helm charts), data simulators, and training pipelines for reproducibility. The experiments were run across multiple cloud sizes to characterize scaling behavior.

Advantages

- Reduced latency and improved freshness via adaptive learned routing compared to static heuristics.
- Ability to express high-level intents (regulatory, clinical priorities) in natural language and have them translated to enforceable constraints.
- Auditability and governance through SAP HANA-backed artifact versioning and provenance tracking.
- Resilience: ensemble and shadowing reduce risk of catastrophic model behavior.
- Cost flexibility: adaptive replication and routing reduce wasted bandwidth and compute by servicing only the most utility-adding streams.

Disadvantages

- Increased system complexity: integrating GenAI, RL policies, and enterprise data platforms increases operational burden.
- Risk of silent model failures if shadowing and canarying are not rigorously enforced.
- GenAI translation errors (misinterpretation of high-level intent) could result in incorrect constraints unless deterministic templating and verification are strict.
- Cost: SAP HANA in-memory usage and high-throughput Kafka setups can be expensive for smaller organizations.
- Regulatory conservatism may require human-in-the-loop approval for certain constraint changes, reducing full automation benefits.

IV. RESULTS AND DISCUSSION

In controlled benchmarks across clinical-AI and financial-risk tasks, our integrated system consistently outperformed baselines on latency and freshness. For sepsis-risk inference, median end-to-end latency dropped from 210 ms (rule-based) to 137 ms (our system), a 34% improvement; 95th-percentile latency reduced by 48 ms on average. Freshness — measured as time from data generation to use in the most recent inference — improved by 28% across clinical streams. For high-frequency financial telemetry, we observed a 56% median latency reduction in burst scenarios due to adaptive replication and prioritized routing. The GenAI orchestration achieved 98.4% accuracy in translating natural-language intents into validated constraints, with failures primarily from highly ambiguous phrasing; deterministic templating and verifier reduced exploitable ambiguity.

Ablation studies showed that constraint masking (applying GenAI-derived hard masks) was essential to guarantee compliance; policies without masking occasionally proposed illegal destinations under heavy reward pressure. Continual learning produced steady improvements but required strict statistical gating to avoid regressions. The system sustained availability and correctness under simulated node failures by rerouting to permitted replicas; in extreme partition scenarios, temporary staleness increased but never violated hard residency constraints.

Operational observations: SAP HANA's in-memory lookups provided sub-millisecond constraint checks but increased memory costs; a hybrid approach caching a subset of high-use entries in the policy node struck an acceptable cost-performance balance. Observability — fine-grained telemetry, distributed tracing, and explainability hooks in the policy network — was crucial for debugging misrouted flows and for compliance audits. Human-in-the-loop interfaces for reviewing GenAI-synthesized constraints were valuable in early deployment phases.

V. CONCLUSION

We demonstrated a practical architecture for GenAI-enabled, neural-network-driven LDDR optimization on a secure Apache–SAP HANA cloud infrastructure tailored to clinical and risk AI workloads. The approach combines intent-aware GenAI translation, learned routing policies, and enterprise-grade streaming and governance primitives to achieve significant latency and freshness gains while preserving regulatory compliance and auditability. Through extensive experiments, we showed the system's ability to improve end-to-end performance, maintain safety under constraints, and adapt to changing conditions.

VI. FUTURE WORK

- Formal verification of GenAI-to-constraint translation pipelines to provide mathematical guarantees on correctness for subsets of intents.
- Explore federated and privacy-preserving model training across hospital boundaries to reduce data movement while enabling stronger local models.
- Extend policy networks to jointly optimize cost, carbon footprint, and latency for sustainability-aware routing.
- Evaluate long-term drift mitigation strategies, including meta-learning and continual-distillation approaches to accelerate safe adaptation.
- User studies with clinicians and risk officers to refine natural-language intent specifications and human-in-the-loop workflows.
- Broader interoperability tests across other in-memory and streaming stacks (e.g., Snowflake, Pulsar) to generalize the blueprint.

REFERENCES

1. Adari, V. K. (2021). Building trust in AI-first banking: Ethical models, explainability, and responsible governance. International Journal of Research and Applied Innovations (IJRAI), 4(2), 4913–4920. https://doi.org/10.15662/IJRAI.2021.0402004
2. Raj, A. A., & Sugumar, R. (2023, May). Multi-Modal Fusion of Deep Learning with CNN based COVID-19 Detection and Classification Combining Chest X-ray Images. In 2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 569-575). IEEE.
3. Sudhan, S. K. H. H., & Kumar, S. S. (2015). An innovative proposal for secure cloud authentication using encrypted biometric authentication scheme. Indian journal of science and technology, 8(35), 1-5.
4. Goriparthi, R. G. (2021). Scalable AI Systems for Real-Time Traffic Prediction and Urban Mobility Management. International Journal of Advanced Engineering Technologies and Innovations, 1(2), 255-278.
5. Mohile, A. (2023). Next-Generation Firewalls: A Performance-Driven Approach to Contextual Threat Prevention. International Journal of Computer Technology and Electronics Communication, 6(1), 6339-6346.
6. Chatterjee, P. (2019). Enterprise Data Lakes for Credit Risk Analytics: An Intelligent Framework for Financial Institutions. Asian Journal of Computer Science Engineering, 4(3), 1-12. https://www.researchgate.net/profile/Pushpalika-Chatterjee/publication/397496748_Enterprise_Data_Lakes_for_Credit_Risk_Analytics_An_Intelligent_Framework_for_Financial_Institutions/links/69133ebec900be105cc0ce55/Enterprise-Data-Lakes-for-Credit-Risk-Analytics-An-Intelligent-Framework-for-Financial-Institutions.pdf
7. Kumar, S. N. P. (2022). Improving Fraud Detection in Credit Card Transactions Using Autoencoders and Deep Neural Networks (Doctoral dissertation, The George Washington University).
8. Anand, P. V., & Anand, L. (2023, December). An Enhanced Breast Cancer Diagnosis using RESNET50. In 2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES) (pp. 1-5). IEEE.
9. Konda, S. K. (2024). AI Integration in Building Data Platforms: Enabling Proactive Fault Detection and Energy Conservation. International Journal of Advanced Research in Computer Science & Technology (IJARCST), 7(3), 10327-10338.

10. Karanjkar, R. (2022). Resiliency Testing in Cloud Infrastructure for Distributed Systems. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 5(4), 7142-7144.

11. Peram, S. (2023). Machine Learning in Wealth Management: Enhancing Investment Strategies through AI. https://www.researchgate.net/profile/Sudhakara-Peram/publication/396293166_Machine_Learning_in_Wealth_Management_Enhancing_Investment_Strategies_through_AI/links/68e5f128ffdca73694b6174e/Machine-Learning-in-Wealth-Management-Enhancing-Investment-Strategies-through-AI.pdf

12. Christadoss, J., Yakkanti, B., & Kunju, S. S. (2023). Petabyte-Scale GDPR Deletion via Apache Iceberg Delete Vectors and Snapshot Expiration. European Journal of Quantum Computing and Intelligent Agents, 7, 66-100.

13. Pasumarthi, A. (2023). Dynamic Repurpose Architecture for SAP Hana Transforming DR Systems into Active Quality Environments without Compromising Resilience. International Journal of Engineering & Extended Technologies Research (IJEETR), 5(2), 6263-6274.

14. Joseph, J. (2023). Trust, but Verify: Audit-ready logging for clinical AI. https://www.researchgate.net/profile/JimmyJoseph9/publication/395305525_Trust_but_Verify_Audit-ready_logging_for_clinical_AI/links/68bbc5046f87c42f3b9011db/Trust-but-Verify-Audit-readylogging-for-clinical-AI.pdf

15. Kotapati, V. B. R., & Yakkanti, B. (2023). Real-Time Analytics Optimization Using Apache Spark Structured Streaming: A Lambda Architecture-based Scala Framework. American Journal of Data Science and Artificial Intelligence Innovations, 3, 86-119.

16. Harish, M., & Selvaraj, S. K. (2023, August). Designing efficient streaming-data processing for intrusion avoidance and detection engines using entity selection and entity attribute approach. In AIP Conference Proceedings (Vol. 2790, No. 1, p. 020021). AIP Publishing LLC.

17. Adari, V. K., Chunduru, V. K., Gonepally, S., Amuda, K. K., & Kumbum, P. K. (2024). Artificial Neural Network in Fibre-Reinforced Polymer Composites using ARAS method. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 7(2), 9801-9806.

18. Kumar, R., Al-Turjman, F., Anand, L., Kumar, A., Magesh, S., Vengatesan, K., ... & Rajesh, M. (2021). Genomic sequence analysis of lung infections using artificial intelligence technique. Interdisciplinary Sciences: Computational Life Sciences, 13(2), 192-200.

19. Kesavan, E. Developing A Testing Maturity Model for Software Test Process Evaluation and Improvement using the DEMATEL Method. https://d1wqtxts1xzle7.cloudfront.net/124509220/Developing_A_Testing_Maturity_Model_for_Software_Test_Process_Evaluation_and_Improvement_using_the_DEMATEL_Method_1-libre.pdf?1757232956=&response-content-disposition=inline%3B+filename%3DDeveloping_A_Testing_Maturity_Model_for.pdf&Expires=1762449739&Signature=WF5l9kUpPuqrSE376hcDC9st4xWv9K9P-OedL8ydfiXp5Np~p0M8dvEvP2-k9NaWjGdfvcw2DoT3X9Fca7PG9-IgxQEoodbyt1rVJ-n2ZHqmuQ2~bMT-tBzSluQmw65jOy7a7PFkFizJEYF6Fz9TLwASEzDBB4gt8HoJtp8NwwrFY-cvrgQHU7x64ab3Cva8hqaS947HBXofRk1~5cGYjdwvAP4E4fotrZxZ~oKwn9Iq8bkobL376q0r7x~LjLXWEE4y~VzKQf8EIgiN3aDI3WkYn08vdDTnEvJMhfWWV-wSPlm0oqp9KFditEDByBQC5eRr6TUnsZwP3a4sc2Lj0Q__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

20. Mani, R., & Sivaraju, P. S. (2024). Optimizing LDDR Costs with Dual-Purpose Hardware and Elastic File Systems: A New Paradigm for NFS-Like High Availability and Synchronization. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 7(1), 9916-9930.

21. Sudha, N., Kumar, S. S., Rengarajan, A., & Rao, K. B. (2021). Scrum Based Scaling Using Agile Method to Test Software Projects Using Artificial Neural Networks for Block Chain. Annals of the Romanian Society for Cell Biology, 25(4), 3711-3727.

22. Balaji, K. V., & Sugumar, R. (2023, December). Harnessing the Power of Machine Learning for Diabetes Risk Assessment: A Promising Approach. In 2023 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI) (pp. 1-6). IEEE.

23. Sudha, N., Kumar, S. S., Rengarajan, A., & Rao, K. B. (2021). Scrum Based Scaling Using Agile Method to Test Software Projects Using Artificial Neural Networks for Block Chain. Annals of the Romanian Society for Cell Biology, 25(4), 3711-3727.