INTERNATIONAL MULTI DISCIPLINARY JOURNAL FOR RESEARCH & DEVELOPMENT
eISSN 2394-6334 Impact factor: 7,854 Volume 13, issue 08 (2026)

AI-Driven Multi-Agent Framework for Resilient and Scalable Event Data Streaming

Aisha Karimova
Department of Artificial Intelligence,
Tashkent Institute of Digital Technologies Uzbekistan

Abstract: The increasing volume, velocity, and heterogeneity of event data require streaming architectures capable of adapting continuously to changing workloads while maintaining low latency, high throughput, and operational resilience. Conventional event-streaming pipelines frequently rely on relatively static resource allocation, routing, and fault-management mechanisms, which can become inefficient when traffic patterns, resource availability, or application priorities change dynamically. This paper proposes an AI-driven multi-agent framework for resilient and scalable event data streaming in which specialized intelligent agents collaboratively perform workload assessment, resource adaptation, fault detection, routing optimization, and streaming control. The theoretical foundation integrates continual-learning principles with distributed agent coordination so that streaming decisions can evolve without requiring complete retraining after every environmental change. Existing continual-learning studies demonstrate the importance of controlling interference, preserving useful knowledge, adapting model parameters, and maintaining task-specific behavior under sequentially changing conditions (Aljundi et al., 2019a; Ahn et al., 2019; Adel et al., 2020). Building on these principles, the proposed framework conceptualizes event streaming as a continuously evolving decision environment rather than a static data-processing problem. The methodology defines cooperating agents for workload monitoring, resource allocation, fault management, and adaptive stream optimization, coordinated through feedback-driven decision cycles. The framework is positioned as an architectural and analytical contribution rather than a claim of experimentally measured performance. Its principal expected outcomes are improved adaptability, reduced disruption during workload changes, better resource utilization, and enhanced resilience under streaming failures. The study further identifies limitations associated with agent coordination overhead, continual-learning instability, computational cost, and the possibility of conflicting local objectives. The resulting framework provides a research foundation for intelligent, adaptive, and scalable event-streaming systems.

Keywords: AI-driven streaming; multi-agent systems; event data; continual learning; resilient computing; adaptive resource allocation; scalable streaming; intelligent routing; fault detection; real-time data processing.

INTRODUCTION

Modern data-intensive applications increasingly operate on continuous streams rather than isolated and independently processed datasets. Event streams generated by distributed services, sensors, transactional platforms, monitoring systems, and interactive applications can change rapidly in volume, composition, and temporal behavior. Consequently, a streaming architecture must address not only data processing but also continuous adaptation to changing operating conditions. A system that performs efficiently under one workload may experience congestion, resource imbalance, or increased failure exposure when event rates or processing requirements change.

This problem becomes particularly important when the streaming environment is heterogeneous and geographically or computationally distributed. Static routing policies and predetermined resource configurations can become inadequate because the optimal processing configuration is dependent on current workload characteristics. The central research problem is therefore how an event-streaming architecture can make distributed decisions dynamically while maintaining resilience and scalability.

The proposed approach addresses this challenge through an AI-driven multi-agent framework. Instead of assigning all intelligence to a single centralized controller, the architecture distributes decision-making among specialized agents. Each agent observes a particular aspect of the streaming environment and contributes to a coordinated control process. This approach is theoretically compatible with continual-learning research, where models must learn from sequentially changing conditions while limiting destructive interference between previously acquired knowledge and new information (Aljundi et al., 2019a).

Continual learning is particularly relevant because event-streaming environments are inherently non-static. Task distributions may change, workloads may shift, and operational states may repeatedly recur. Research on adaptive weights, uncertainty-based learning, separated classification strategies, and memory-aware mechanisms demonstrates different approaches to maintaining useful knowledge while adapting to new conditions (Adel et al., 2020; Ahn et al., 2019; Ahn et al., 2021; Aljundi et al., 2018). These ideas provide a theoretical basis for developing streaming agents that adapt incrementally rather than treating every workload change as an entirely new problem.

The principal objective of this research is to formulate an integrated framework in which intelligent agents collaboratively optimize event-streaming operations. The framework focuses on four architectural concerns: workload observation, resource allocation, fault management, and adaptive stream control. The study also examines how continual-learning mechanisms can support these agents and identifies the trade-offs associated with distributed intelligent control.

The significance of the proposed framework lies in its attempt to connect two traditionally distinct perspectives: adaptive machine learning and resilient event-stream processing. Rather than considering learning as an isolated analytics function, the proposed architecture treats learning as an operational control mechanism capable of influencing routing, resource utilization, and fault recovery. This conceptual integration is consistent with the broader objective of AI-based multi-agent streaming optimization described by Reddy et al. (2026).

LITERATURE REVIEW

The supplied literature is predominantly concerned with continual learning, domain adaptation, few-shot incremental learning, and mechanisms for retaining or adapting learned representations. Although these studies do not directly establish a complete event-streaming architecture, they provide important theoretical foundations for intelligent streaming agents.

Achille et al. (2018) investigate lifelong disentangled representation learning through cross-domain latent relationships. Their work is relevant to event streaming because continuously arriving data may exhibit changing distributions and domain characteristics. An intelligent streaming agent must therefore distinguish information that represents persistent operational structure from information that reflects temporary workload conditions. A representation-learning perspective can support this distinction by enabling an agent to retain transferable characteristics while adapting to new data regimes.

Aljundi et al. (2018) introduce memory-aware synapses for learning what should not be forgotten. This principle is important for resilient streaming control because adaptation without retention can cause previously effective operational policies to disappear after workload changes. In a multi-agent streaming framework, memory-aware learning can therefore be interpreted as a mechanism for preserving knowledge about stable routing patterns, resource requirements, or failure responses.

The problem of interference is addressed more explicitly by Aljundi et al. (2019a), whose work on online continual learning with maximally interfered retrieval emphasizes the difficulty of learning continuously when new information can conflict with previously acquired knowledge. Streaming systems face an analogous problem: a workload change may require a new resource policy that conflicts with a previously optimized configuration. A streaming agent consequently requires mechanisms that can adapt without destabilizing established behavior.

Ahn et al. (2019) examine uncertainty-based continual learning with adaptive regularization. The uncertainty perspective is particularly useful for event-streaming environments because operational decisions are rarely made under complete information. An agent may observe uncertain workload forecasts, incomplete resource states, or ambiguous failure signals. Incorporating uncertainty into decision-making can therefore reduce aggressive adaptation when the evidence for change is weak.

Ahn et al. (2021) propose separated softmax for incremental learning. The underlying importance for the proposed architecture is the separation of knowledge associated with incrementally emerging conditions. In an event-streaming environment, different traffic regimes or application classes may require distinct decision boundaries. Separating these conditions can help prevent newly observed operational patterns from disproportionately influencing decisions associated with established patterns.

Adel et al. (2020) address continual learning through adaptive weights. This concept directly supports the proposed architecture because adaptive parameterization allows agents to modify their decision behavior according to changing environmental conditions. Rather than permanently fixing resource or routing policies, an adaptive agent can adjust decision parameters as the streaming environment evolves.

Ajakan et al. (2014) examine domain-adversarial neural networks. Domain adaptation is relevant because event streams may originate from heterogeneous sources whose statistical properties differ. A streaming controller that learns directly from heterogeneous data may incorrectly interpret source-specific variation as operational change. Domain-invariant representations can conceptually reduce this problem by encouraging agents to learn features that remain useful across different operating domains.

Akyurek et al. (2021) study subspace regularization for few-shot class-incremental learning. The work suggests another useful perspective: new operational states may sometimes appear with limited observations. A streaming agent should therefore be capable of adapting to emerging conditions without requiring large quantities of newly labeled data. Subspace-based adaptation provides a conceptual mechanism for limiting disruptive changes while incorporating new information.

Collectively, the literature identifies three central principles for the proposed framework. First, adaptive learning is necessary when the environment changes continuously (Adel et al., 2020). Second, knowledge retention and interference control are necessary when adaptation occurs sequentially (Aljundi et al., 2018; Aljundi et al., 2019a). Third, uncertainty and domain variation must be explicitly considered because changing data distributions can otherwise produce unstable decisions (Ajakan et al., 2014; Ahn et al., 2019).

The primary research gap is that these learning mechanisms are generally considered at the model or representation level rather than as components of a coordinated event-streaming control architecture. The proposed research addresses this gap by positioning continual-learning mechanisms within specialized agents responsible for operational streaming decisions. The framework therefore extends the conceptual role of adaptive learning from prediction toward distributed stream management. This positioning is aligned with the AI-driven multi-agent streaming perspective proposed by Reddy et al. (2026).

METHODOLOGY

3.1 Research Design and Architectural Principle

This research adopts a framework-development methodology. Rather than claiming experimentally validated numerical improvements, the study develops an analytically grounded architecture based exclusively on the supplied literature and the stated multi-agent streaming objective. The framework treats event streaming as a partially changing decision environment in which data arrival, computational demand, resource availability, and failure conditions evolve over time.

The central design principle is specialization with coordination. Instead of one monolithic AI model controlling the entire streaming pipeline, multiple agents are assigned distinct responsibilities. This decomposition reduces the complexity of individual decision problems and allows each agent to maintain a specialized learning state.

The conceptual framework consists of an event-ingestion layer, observation layer, intelligent-agent layer, coordination layer, execution layer, and feedback loop. Events enter through the ingestion layer and are characterized according to rate, priority, source, processing requirements, and current system conditions. The observation layer collects operational signals such as queue growth, processing delay, resource utilization, and failure indicators. These observations are supplied to specialized agents.

3.2 Workload Intelligence Agent

The workload intelligence agent continuously estimates the current and near-future state of event traffic. Its primary task is to identify workload regimes rather than merely count incoming events. For example, a sudden increase in event volume may indicate a temporary burst, while a persistent increase may indicate a structural workload transition.

Continual-learning principles are useful because the workload distribution can change over time. Adaptive weights provide a mechanism through which the agent can progressively update its decision function (Adel et al., 2020). However, uncontrolled adaptation can cause instability. Memory-aware learning can therefore be incorporated to preserve parameters associated with historically reliable workload patterns (Aljundi et al., 2018).

The agent should also estimate uncertainty. When a workload prediction is highly uncertain, the framework should avoid making aggressive resource changes unless supporting evidence is available. This principle follows the relevance of uncertainty-aware adaptation demonstrated by Ahn et al. (2019).

3.3 Resource Allocation Agent

The resource allocation agent translates workload assessments into computational decisions. Its objective is to match available processing capacity with the estimated demand of active event streams.

A simplified resource allocation objective can be represented as:

[

$$R^*=\arg\min_R [\alpha L(R)+\beta C(R)+\gamma U(R)+\delta F(R)]$$

]

where (R) represents a candidate resource configuration, (L) represents latency, (C) represents computational cost, (U) represents resource imbalance, and (F) represents estimated failure exposure. The coefficients $(\alpha,\beta,\gamma,\delta)$ express application-specific priorities.

The important characteristic is that the allocation policy should not remain fixed. As workload states evolve, the agent updates its decision policy. Adaptive learning provides the mechanism for incremental adjustment, while regularization and knowledge retention reduce the risk of excessive policy drift (Ahn et al., 2019; Aljundi et al., 2019a).

3.4 Fault Detection and Recovery Agent

The fault agent monitors operational anomalies and determines whether an observed deviation represents a transient fluctuation or a meaningful failure condition. Examples include unexpectedly increasing processing latency, queue accumulation, unavailable processing nodes, or repeated stream interruptions.

A resilient design requires more than detecting failures. The agent must connect detection to an appropriate response. Possible responses include rerouting event partitions, reallocating resources, reducing processing complexity, or transferring workloads to alternative agents.

The continual-learning perspective is important because failure behavior may change over time. A failure signature learned under one configuration may not remain identical after architectural or workload changes. Therefore, the fault agent should retain previously meaningful patterns while adapting to newly observed conditions. This requirement reflects the broader stability-plasticity challenge addressed by memory-aware and online continual-learning approaches (Aljundi et al., 2018; Aljundi et al., 2019a).

3.5 Stream Optimization Agent

The stream optimization agent acts as the final operational controller for event movement and processing. It integrates information from the workload, resource, and fault agents. Its objective is to select an operational configuration that balances throughput, latency, resource efficiency, and resilience.

The agent can be conceptualized as a policy function:

[

$$a_t=\pi(s_t,\theta_t)$$

]

where (s_t) represents the current streaming state, (θ_t) represents the continuously adapted model parameters, and (a_t) represents the selected streaming action.

The principal advantage of this formulation is adaptability. The policy is not permanently associated with one static workload. Instead, it changes as the system observes new states. However, continual adaptation must be constrained to prevent catastrophic behavioral changes. Online learning with interference-aware retrieval and adaptive regularization provides relevant theoretical mechanisms for this purpose (Aljundi et al., 2019a; Ahn et al., 2019).

3.6 Multi-Agent Coordination

Individual agents cannot operate independently because their objectives may conflict. A resource agent may seek additional capacity while a cost-sensitive controller may seek to minimize resource consumption. Similarly, a fault agent may recommend rerouting while the optimization agent may prioritize latency.

The coordination layer therefore establishes a shared operational state and prioritization policy. Agent recommendations are evaluated according to global streaming objectives rather than isolated local objectives. The architecture can employ weighted decision aggregation:

[

$$A_t=\arg\max_a \sum_{i=1}^{n} w_i Q_i(s_t,a)$$

]

where (Q_i) represents the recommendation quality of agent (i), (w_i) represents its current importance, and (A_t) is the selected global action.

The weights can themselves be adaptive. This is conceptually consistent with adaptive-weight continual-learning approaches (Adel et al., 2020). Such a mechanism allows the framework to prioritize fault recovery during instability while emphasizing efficiency during normal operation.

3.7 Continual-Learning Feedback Cycle

The framework operates through a repeated cycle:

observe → estimate → coordinate → act → evaluate → retain/adapt → observe again.

After each action, the system evaluates its operational consequences. Successful decisions strengthen the associated policy, whereas unsuccessful decisions contribute corrective information. This process avoids treating learning as a one-time training stage.

Cross-domain and representation-learning concepts are relevant when event streams originate from different sources or application domains (Achille et al., 2018; Ajakan et al., 2014). The framework should preserve domain-general operational features while preventing source-specific characteristics from dominating global policies.

Few-shot incremental learning is also relevant when new event categories or workload states emerge with limited historical data. Subspace regularization provides a conceptual basis for adapting to such new conditions without completely replacing established representations (Akyurek et al., 2021).

3.8 Hypothetical Operational Scenario

Consider an event-streaming platform experiencing a sudden increase in events from multiple sources. The workload agent detects the change and estimates whether the increase is transient or persistent. The resource agent identifies available processing capacity and recommends additional allocation. Simultaneously, the fault agent detects that one processing path has developed abnormal latency. The stream optimization agent reroutes part of the workload while the coordination layer prioritizes continuity over short-term resource minimization.

After the workload stabilizes, the agents evaluate the effectiveness of their decisions. If the observed pattern recurs, the system can reuse the learned operational knowledge. If the workload characteristics differ substantially, adaptive learning modifies the policy. Memory-aware mechanisms reduce the probability that this adaptation will erase previously effective behavior (Aljundi et al., 2018).

RESULTS

The framework analysis indicates that the primary contribution of an AI-driven multi-agent architecture is not simply the introduction of multiple machine-learning models but the transformation of streaming management into a continuous adaptive control problem. The proposed architecture creates a functional separation between workload understanding, resource allocation, fault management, and stream optimization. Such separation can reduce the decision complexity assigned to individual agents while allowing their outputs to be coordinated at the system level.

The first expected finding is improved adaptability under workload variation. Static streaming policies implicitly assume that historical operating conditions remain representative of future conditions. This assumption becomes increasingly weak when event rates and source characteristics evolve. Adaptive weights and continual-learning mechanisms provide a theoretical basis for modifying policies without rebuilding the entire model from the beginning (Adel et al., 2020). Consequently, the proposed architecture should be more responsive to changing workload states than a strictly static control strategy.

The second finding concerns resilience. A dedicated fault agent allows abnormal conditions to be recognized as operational events rather than merely statistical anomalies. Its integration with the resource and optimization agents enables detection to be followed by corrective action. Memory-aware learning is particularly important because previously successful recovery policies may remain valuable even after new failure patterns are introduced (Aljundi et al., 2018).

The third finding is improved knowledge retention during continuous adaptation. The literature indicates that sequential learning creates a tension between plasticity and stability. Online learning can incorporate new information but may also introduce interference with earlier knowledge (Aljundi et al., 2019a). The proposed framework addresses this tension by combining adaptive parameter modification with retention-oriented mechanisms. This creates a more suitable learning structure for streaming environments in which old and new workload states coexist.

A fourth finding concerns uncertainty. Streaming decisions are frequently made under incomplete information, making confidence-aware adaptation important. Uncertainty-based regularization provides a theoretical basis for reducing excessive reactions to ambiguous environmental changes (Ahn et al., 2019). Therefore, the framework should distinguish between statistically meaningful workload transitions and short-lived fluctuations.

The framework also indicates that domain variability must be treated as a first-class architectural concern. When multiple data sources exhibit different distributions, direct model adaptation may confuse source differences with operational changes. Domain-adaptive learning can help agents identify more transferable characteristics across heterogeneous environments (Ajakan et al., 2014).

However, the findings also expose significant trade-offs. Multi-agent coordination introduces communication and decision overhead. Continual learning introduces additional computational requirements, while excessive specialization can produce conflicting recommendations. Therefore, resilience and scalability cannot be assumed merely because AI agents are present. Their benefits depend on coordination quality, learning stability, and the computational cost of maintaining intelligent control.

Accordingly, the results should be interpreted as analytical framework findings rather than experimentally measured performance. The architecture establishes mechanisms through which resilience and scalability may be improved, but quantitative validation requires implementation and controlled experiments. The framework is conceptually consistent with the objective of AI-based multi-agent optimization for event streaming described by Reddy et al. (2026).

DISCUSSION

The proposed framework contributes theoretically by repositioning continual learning as an operational mechanism for streaming-system control. Traditional interpretations of continual learning focus primarily on preserving model competence while acquiring new knowledge. In the proposed architecture, the same principle is extended to system-level decisions, where learned knowledge influences routing, allocation, fault response, and stream optimization. This extension is important because an event-streaming environment changes continuously, making static learning assumptions insufficient.

The literature supports this theoretical position from several complementary directions. Adaptive weights provide plasticity, allowing decision functions to change as conditions evolve (Adel et al., 2020). Memory-aware mechanisms provide stability by identifying information that should be retained (Aljundi et al., 2018). Online interference-aware learning highlights the difficulty of simultaneously learning recent patterns and preserving historical competence (Aljundi et al., 2019a). These mechanisms collectively suggest that a resilient streaming controller requires both adaptation and retention rather than maximizing either property independently.

A major practical implication is that agent specialization can make complex streaming control more manageable. A workload agent does not need to solve fault recovery, while a fault agent does not need to independently optimize every resource decision. This decomposition can improve interpretability because operational actions can be traced to particular functional responsibilities. However, decomposition also creates coordination costs. If agents receive inconsistent information or optimize incompatible objectives, the distributed architecture may become less efficient than a centralized controller.

Uncertainty represents another critical trade-off. An aggressive adaptive system may respond rapidly but risk overreacting to noise. A conservative system may preserve stability but respond too slowly to genuine workload transitions. The uncertainty-based perspective of Ahn et al. (2019) suggests that adaptation intensity should depend on confidence in the observed state. This principle can be integrated into the coordination layer by increasing the influence of adaptive actions when confidence is high and emphasizing historical policies when uncertainty is high.

Domain variation creates an additional challenge. Cross-domain representation learning and domain-adversarial approaches indicate that models may need mechanisms for distinguishing transferable structure from domain-specific characteristics (Achille et al., 2018; Ajakan et al., 2014). In event streaming, this distinction can affect whether a new traffic pattern should trigger global policy adaptation or only local adjustment within one source domain.

The proposed architecture also exposes a scalability paradox. Adding more agents can distribute computational responsibility, but it can simultaneously increase coordination complexity. Therefore, scalability should be evaluated at both the data-processing level and the intelligence-control level. A system may process larger event volumes successfully while its coordination mechanism becomes a bottleneck. This observation implies that agent communication frequency, decision aggregation, and model-update schedules must be treated as explicit architectural parameters.

Another limitation concerns emerging states with insufficient observations. Few-shot incremental learning provides a useful theoretical perspective because new workload classes may appear before enough data exist for robust adaptation (Akyurek et al., 2021). Consequently, the framework should avoid treating every new state as fully trustworthy. Incremental adaptation should be constrained by uncertainty, historical knowledge, and the expected operational consequences of incorrect decisions.

Finally, the framework should be distinguished from an experimentally validated streaming implementation. Its contribution is architectural and theoretical: it establishes how adaptive learning principles can be organized into a coordinated multi-agent control structure. Empirical validation remains necessary to measure latency, throughput, fault-recovery time, resource efficiency, coordination overhead, and learning stability under controlled workloads. The direction is consistent with the broader AI-driven multi-agent streaming objective identified by Reddy et al. (2026).

CONCLUSION

This study presented an AI-driven multi-agent framework for resilient and scalable event data streaming. The framework conceptualizes streaming as a continuously changing operational environment in which intelligent agents must observe workloads, allocate resources, detect faults, optimize stream decisions, and learn from feedback. Its theoretical foundation is derived from continual-learning research concerning adaptive weights, uncertainty management, domain adaptation, memory retention, interference control, and incremental learning.

The principal contribution is the integration of these learning principles into a coordinated multi-agent architecture. Rather than using AI solely for prediction, the proposed framework positions intelligent models as active components of streaming-system control. This enables a transition from static resource and routing policies toward adaptive operational decision-making.

The analysis indicates that resilience depends not only on detecting failures but also on retaining effective recovery knowledge and adapting to new failure patterns. Similarly, scalability requires more than adding resources; it requires intelligent allocation and coordination as workload conditions evolve. The framework therefore treats adaptability, resilience, and scalability as interdependent properties.

Several limitations remain. Continual learning may introduce computational overhead, multi-agent coordination can create conflicting objectives, and adaptation may become unstable under highly uncertain conditions. In addition, the framework has not been presented as an experimentally validated implementation. Future research should implement the architecture and evaluate it using realistic event-streaming workloads, measuring latency, throughput, fault-recovery time, resource utilization, coordination overhead, and learning stability. Comparative evaluation against centralized and non-adaptive streaming controllers would further establish the practical value of the proposed approach.

Overall, the framework provides a theoretically grounded foundation for intelligent event-streaming infrastructures in which learning, resource management, and resilience are treated as components of a unified adaptive control system. Its central proposition is that continual learning and multi-agent coordination can jointly support streaming architectures capable of responding to changing workloads while preserving previously acquired operational knowledge.

REFERENCES

1. Abati, D., Tomczak, J., Blankevoort, T., Calderara, S., Cucchiara, R., & Bejnordi, B. (2020). Conditional channel gated networks for task-aware continual learning. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

2. Achille, A., Eccles, T., Matthey, L., Burgess, C., Watters, N., Lerchner, A., & Higgins, I. (2018).Life-long disentangled representation learning with cross-domain latent homologies .Advances in Neural Information Processing Systems (NIPS).

3. Adel, T., Zhao, H., & Turner, R. (2020). Continual learning with adaptive weights (CLAW).International Conference on Learning Representations (ICLR).

4. Ahn, H., Kwak, J., Lim, S., Bang, H., Kim, H., & Moon, T. (2021). SS-IL: Separated softmax forincremental learning. Proceedings of the IEEE/CVF International Conference on Computer Vision.

5. Ahn, H., Lee, D., Cha, S., & Moon, T. (2019). Uncertainty-based continual learning with adaptivere gularization. Advances in Neural Information Processing Systems (NeurIPS).

6. Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., & Marchand, M. (2014). Domain adversarialneural networks. CoRR, abs/1412.4446.

7. Akyurek, A., Akyurek, E., Wijaya, D., & Andreas, J. (2021). Subspace regularizers for few-shotclass incremental learning.arXiv preprint arXiv:2110.07059.

8. Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., & Tuytelaars, T. (2018). Memory awaresynapses: Learning what (not) to forget. European Conference on Computer Vision (ECCV).

9. Aljundi, R., Caccia, L., Belilovsky, E., Caccia, M., Lin, M., Charlin, L., & Tuytelaars, T. (2019a).Online continual learning with maximally interfered retrieval. Advances in Neural Information Processing Systems (NeurIPS).

10. R. Reddy, P. Udayaraju, K. K. Goyal, S. C. R. Vudem, R. Sayana and V. Gummadi, "Creating an AI-based Multi-Agent Model for Optimized Data Streaming with Improved Resiliency and Scalability for Event Streaming," 2026 6th International Conference on Image Processing and Capsule Networks (ICIPCN), Dhulikhel, Nepal, 2026, pp. 1372-1379, doi: 10.1109/ICIPCN67432.2026.11438445.

11. K. Ramamurthy, N. Bellamkonda and N. Amanmadov, "ScalePulse: Combinatorial Llm Framework for Scalability Constraint," SoutheastCon 2026, Huntsville, AL, USA, 2026, pp. 1-6, doi: 10.1109/SoutheastCon63549.2026.11476075

12. Philip, P. G. (2024). Artificial Intelligence-Driven Project Risk Prediction Models: Enhancing Decision-Making Accuracy in Large-Scale Infrastructure Projects . American Journal of Technology, 3(1), 52–69. https://doi.org/10.58425/ajt.v3i1.571

13. Ramaswamy, K., Kodela, S., Pal, M., & Chauhan, R. (2026, March). Smart Cloud Optimization Platform Using Multi-Agent AI, Trust Analytics, and Energy-Aware Task Scheduling. In 2026 Innovations in Machine, Engineering, and Digital Conference (IMED) (pp. 1-6). IEEE.