American Journal of Applied Science and Technology

VOLUME Vol.06 Issue 08 2026
PAGE NO. 145-152
DOI10.37547/ajast/Volume06Issue08-10

Adaptive Multi-Agent AI Framework for Real-Time Data Streaming with Enhanced Scalability and Resilience

Nethmi Perera

Department of Artificial Intelligence, Colombo Institute of Computing Sri Lanka

Kasun Fernando

Department of Intelligent Systems, Kandy Institute of Technology Sri Lanka

Received: 28 June 2026; **Accepted:** 22 July 2026; **Published:** 20 August 2026

Abstract: Real-time data streaming systems increasingly operate under highly variable workloads, heterogeneous data sources, latency constraints, and frequent service disruptions. Conventional stream-processing architectures generally depend on predefined routing, static resource allocation, and centralized coordination, which can limit their ability to adapt when event rates, computational requirements, or infrastructure conditions change rapidly. This paper proposes an Adaptive Multi-Agent AI Framework for Real-Time Data Streaming with Enhanced Scalability and Resilience, in which autonomous AI agents collaboratively perform stream monitoring, workload classification, task allocation, resource adaptation, anomaly detection, and recovery. The theoretical foundation combines multi-agent coordination with contextual representation, long-document processing, memory management, and adaptive decision-making. Prior work on aspect-controllable summarization demonstrates the value of controlling computational objectives according to task requirements, while studies of coreference, lexical chains, and entity-based coherence emphasize the importance of preserving relationships across distributed information units (Amplayo, Angelidis, & Lapata, 2021; Baldwin & Morton, 1998; Barzilay & Elhadad, 1997; Barzilay & Lapata, 2005). Long-context language modeling further motivates mechanisms capable of retaining relevant information over extended streaming windows (Beltagy, Peters, & Cohan, 2020). The proposed framework extends these principles to adaptive streaming environments and aligns with recent multi-agent event-streaming research emphasizing resiliency and scalability (Reddy et al., 2026). Analytical findings indicate that decentralized agent specialization, shared contextual state, adaptive workload redistribution, and failure-aware coordination can provide a stronger basis for resilient streaming than static pipelines. The paper also identifies trade-offs involving coordination overhead, state consistency, model complexity, and resource consumption.

Keywords: Multi-Agent AI, Real-Time Data Streaming, Adaptive Computing, Scalability, Resilience, Event Processing, Intelligent Workload Allocation, Distributed AI, Stream Processing

1. INTRODUCTION

The Real-time data streaming has become a fundamental computational paradigm for applications in which information must be processed continuously rather than accumulated for periodic batch analysis. Event-driven applications such as intelligent monitoring, distributed analytics, industrial automation, financial event processing, and operational decision systems may receive streams whose volume, velocity, and semantic characteristics change continuously. Such environments create a fundamental systems challenge: an architecture optimized for one workload condition may become inefficient or unreliable when event rates increase, processing requirements change, or individual services fail.

Traditional stream-processing architectures commonly divide processing into predefined stages. Although this modularity improves operational clarity, static task allocation can create bottlenecks when workload distributions become uneven. A single overloaded component can increase end-to-end latency even when other computational resources remain underutilized. Similarly, centralized decision-making can become a single point of coordination failure. The problem is therefore not merely to process events quickly, but to determine which processing action should occur, where it should occur, and how the system should react when operating conditions change.

Recent work on AI-based multi-agent streaming provides a direct motivation for combining autonomous decision-making with event-streaming infrastructure. Reddy et al. (2026) describe an AI-based multi-agent model directed toward optimized data streaming with improved resiliency and scalability. The present framework builds upon this direction by emphasizing adaptive agent specialization, contextual state management, dynamic workload redistribution, and failure-aware coordination rather than treating streaming as a fixed sequence of processing stages.

The conceptual foundation can also be derived from research outside conventional stream-processing systems. Aspect-controllable summarization demonstrates that an intelligent system can modify its processing behavior according to a specified objective or aspect (Amplayo, Angelidis, & Lapata, 2021). Dynamic coreference-based summarization illustrates the importance of maintaining relationships among information units during processing (Baldwin & Morton, 1998), while lexical-chain and entity-based approaches demonstrate that coherent interpretation depends on preserving relationships across distributed textual elements (Barzilay & Elhadad, 1997; Barzilay & Lapata, 2005). In a streaming environment, analogous relationships exist among events, sessions, entities, services, and processing states.

This paper therefore proposes a conceptual framework in which autonomous agents observe streaming conditions, maintain contextual information, coordinate processing decisions, and dynamically adapt resource utilization. The principal objective is to establish a research-oriented architecture that improves scalability and resilience while preserving contextual consistency. The scope is primarily architectural and methodological; the paper does not claim results from a physical benchmark deployment.

2. LITERATURE REVIEW

The supplied literature provides several complementary theoretical foundations for an adaptive multi-agent streaming architecture. Although the references primarily concern language processing, summarization, contextual representation, and memory, their underlying computational principles are relevant to intelligent streaming systems.

Amplayo, Angelidis, and Lapata (2021) investigate aspect-controllable opinion summarization, demonstrating the importance of controlling an intelligent processing system according to a particular objective. In a streaming architecture, an equivalent principle can be applied through goal-oriented agents. Instead of applying identical processing to every event, agents can dynamically prioritize objectives such as low latency, high accuracy, anomaly sensitivity, or resource efficiency. The critical implication is that adaptability should operate at the level of computational objectives rather than merely at the level of hardware scaling.

Baldwin and Morton (1998) examine dynamic coreference-based summarization. Their work highlights the importance of identifying relationships between information units before generating an integrated representation. This principle is significant for streaming because individual events rarely exist independently. Events may belong to the same transaction, user session, device, process, or operational incident. An adaptive agent architecture therefore requires mechanisms for associating temporally distributed events with persistent contextual entities.

The work of Barzilay and Elhadad (1997) on lexical chains similarly establishes the value of identifying semantic relationships distributed across a document. In real-time streams, analogous relationships can be interpreted as chains of related events. For example, a sequence of authentication failures, unusual access events, and subsequent privilege changes may represent a single operational pattern even though the events occur at different times. An agent capable of identifying such chains can make more informed decisions than one operating exclusively on isolated events.

Barzilay and Lapata (2005) introduce an entity-based approach to modeling local coherence. The theoretical importance of this work for streaming lies in its emphasis on maintaining consistent relationships between entities as information evolves. A multi-agent stream-processing system can apply a similar principle through shared state representations in which agents maintain relationships among events, entities, and processing tasks.

Beltagy, Lo, and Cohan (2019) introduce SciBERT, demonstrating the value of domain-oriented pretrained language representations. For streaming systems, this suggests that agent intelligence should be sensitive to domain-specific semantics rather than relying exclusively on generic representations. A healthcare monitoring stream, industrial telemetry stream, and financial transaction stream may require substantially different semantic interpretations.

Beltagy, Peters, and Cohan (2020) propose Longformer for long-document processing. Its importance to the proposed architecture is the recognition that useful computational decisions may require information extending beyond short contextual windows. A streaming agent similarly needs mechanisms for retaining relevant historical context without continuously processing the complete event history. Selective contextual memory can therefore reduce computational cost while preserving decision-relevant information.

Bengio (2017) discusses a consciousness prior and introduces a theoretical perspective concerning attention, abstraction, and internal representation. Although not a streaming architecture, this work motivates the broader idea that intelligent systems benefit from mechanisms that selectively represent information relevant to current decision processes. In the proposed framework, this principle is operationalized through attention-like prioritization of streaming events and agent states.

Berman, Jonides, and Lewis (2009) examine decay in verbal short-term memory. Their findings are conceptually relevant to streaming architectures because indefinite retention of all historical information is computationally impractical. An intelligent stream-processing system therefore requires mechanisms for determining what information should remain active, what should be compressed, and what can be discarded.

Collectively, the literature identifies a gap between context-aware intelligent processing and adaptive multi-agent real-time streaming. Existing references provide foundations for controllability, relationship modeling, semantic coherence, long-context processing, and selective memory, while Reddy et al. (2026) explicitly connect multi-agent AI with optimized event streaming and resiliency. The proposed framework integrates these principles into a unified architecture focused on continuous adaptation.

3. METHODOLOGY

3.1 Proposed Architectural Model

The proposed framework consists of five conceptual layers: the event-ingestion layer, perception and monitoring layer, agent-coordination layer, adaptive execution layer, and resilience layer.

The event-ingestion layer receives heterogeneous streams from sensors, applications, services, databases, or external event producers. Events are normalized into a common representation containing temporal, semantic, source, priority, and processing metadata. This separation allows downstream agents to operate independently of source-specific formats.

The perception and monitoring layer continuously observes event velocity, queue length, processing latency, resource utilization, error frequency, and event characteristics. Rather than treating these metrics as static operational statistics, the framework converts them into decision variables for autonomous agents.

The agent-coordination layer contains specialized agents. A stream-monitoring agent identifies workload changes; a classification agent determines event type and priority; a routing agent selects appropriate processing destinations; a resource-management agent evaluates scaling requirements; a context agent maintains relevant historical relationships; and a resilience agent detects failures and initiates recovery actions. These agents do not need to perform identical computations. Their specialization reduces the cognitive and computational burden imposed on individual components.

The adaptive execution layer transforms agent decisions into concrete stream-processing actions. Depending on workload conditions, tasks may be replicated, migrated, reordered, compressed, delayed, or redistributed. The architecture therefore replaces a fixed pipeline with a dynamically configurable computational graph.

The resilience layer monitors service health and evaluates whether alternative agents or processing nodes can assume failed responsibilities. Reddy et al. (2026) emphasize resiliency and scalability in multi-agent event-streaming environments, and the present framework extends this principle through explicit failure-aware task reassignment.

3.2 Agent State and Context Management

A central methodological issue is the management of state. Streaming decisions frequently depend on historical information, but storing every event indefinitely creates excessive memory and computational requirements. The framework consequently separates state into short-term, contextual, and persistent representations.

Short-term state contains recent events required for immediate decisions. Contextual state stores relationships among events, entities, sessions, and ongoing processes. Persistent state contains compressed summaries or long-term patterns that may influence future decisions. This structure is conceptually consistent with the need to preserve relevant contextual information while avoiding unnecessary retention suggested by research on memory and long-context processing (Berman, Jonides, & Lewis, 2009; Beltagy, Peters, & Cohan, 2020).

The context agent assigns relevance scores to historical information. Events strongly associated with current processing tasks receive higher retention priority, whereas redundant information can be compressed or removed. Entity-based coherence provides a theoretical basis for maintaining relationships among events that belong to the same underlying entity or process (Barzilay & Lapata, 2005).

3.3 Adaptive Workload Allocation

Workload allocation is treated as a continuous optimization problem rather than a one-time scheduling decision. Let the system state at time (t) be represented by:

$$S_t = \{L_t, Q_t, R_t, E_t, C_t\}$$

where (L_t) represents latency, (Q_t) queue pressure, (R_t) resource utilization, (E_t) error conditions, and (C_t) contextual workload characteristics.

Each agent evaluates the current state and proposes an action (A_t). The objective can be conceptualized as:

$$A_t^* = \arg\min_{A_t} [\alpha L_t + \beta Q_t + \gamma R_t + \delta E_t]$$

where the coefficients represent application-specific priorities. This formulation allows the system to favor latency-sensitive execution in one environment and resource efficiency or reliability in another.

The controllability principle identified by Amplayo, Angelidis, and Lapata (2021) supports this adaptive objective-driven design. The system does not assume that one processing strategy is universally optimal; instead, the strategy is conditioned on operational requirements.

3.4 Semantic Event Relationship Modeling

A major limitation of purely metric-driven stream processors is that they can observe high traffic without understanding why events are related. The proposed framework addresses this through relationship-aware agents.

For example, consider a distributed application producing a sequence of timeout events, authentication errors, repeated requests, and service restarts. A metric-only system may treat these as independent anomalies. A context-aware system can identify them as a connected event chain. The lexical-chain perspective of Barzilay and Elhadad (1997) and the coreference perspective of Baldwin and Morton (1998) provide theoretical motivation for this relationship-oriented processing.

The framework can therefore maintain an event graph:

$$G_t=(V_t,E_t)$$

where (V_t) represents events or entities and (E_t)

represents temporal, semantic, causal, or operational relationships. Agents can query this graph when determining whether a new event represents an isolated condition or part of an evolving pattern.

3.5 Resilience and Failure Recovery

Resilience is implemented through continuous health assessment and decentralized recovery. Each agent periodically communicates health information to the coordination mechanism. If an agent becomes unavailable, the system identifies another agent capable of executing the same task or activates a degraded processing mode.

A three-stage recovery strategy is proposed. First, the system detects the failure. Second, it isolates the affected task or service to prevent cascading effects. Third, it redistributes responsibility to available agents. Contextual state is transferred using compact state representations rather than replaying the entire event history.

This design is particularly important for high-volume environments because centralized recovery can increase downtime. Multi-agent coordination distributes decision-making responsibility and allows local failures to be addressed without requiring complete system interruption.

3.6 Scalability Mechanism

Scalability is addressed through horizontal agent replication and workload partitioning. When incoming event rates increase, additional processing agents can be instantiated for overloaded functions. When demand decreases, redundant agents can be removed.

However, indiscriminate replication may increase coordination overhead. The proposed system therefore uses threshold-based or predictive scaling. An agent is replicated only when expected performance degradation exceeds an application-specific tolerance. Domain-aware representations, as motivated by SciBERT (Beltagy, Lo, & Cohan, 2019), can additionally support more accurate classification and workload partitioning in specialized streams.

Predictive forecasting can further support adaptive resource allocation by anticipating changes in workload and demand. Vollem et al. (2026) demonstrate the usefulness of combining statistical and deep learning techniques for time-series forecasting, suggesting that predictive models can provide valuable information for proactive decision-making in dynamic computational environments.

Predictive artificial intelligence and machine learning approaches can also support proactive decision-making in dynamic operational environments. Philip (2026) highlights the application of AI and machine learning for project schedule forecasting and time-control, demonstrating the relevance of predictive models for anticipating future conditions and improving planning decisions.

4. RESULTS

The proposed framework produces several analytical findings regarding the relationship between multi-agent intelligence, scalability, and resilience. Because the paper is architectural rather than an empirical benchmark study, these findings represent theoretically derived outcomes rather than measured experimental performance.

The first finding is that agent specialization can reduce the rigidity of conventional streaming pipelines. In a static pipeline, each processing component performs a predefined function regardless of changing workload conditions. The proposed architecture instead permits monitoring, routing, context, resource, and resilience agents to make specialized decisions. This arrangement is consistent with the broader multi-agent streaming direction described by Reddy et al. (2026), while adding explicit mechanisms for contextual reasoning and state adaptation.

The second finding concerns scalability under heterogeneous workloads. Replication of the entire pipeline is inefficient when only one stage becomes overloaded. Specialized agents enable selective scaling. For instance, if semantic classification becomes computationally intensive while ingestion remains stable, classification agents can be replicated independently. This potentially improves resource utilization because computational capacity follows workload concentration rather than being allocated uniformly.

The third finding is that context-aware processing can improve the quality of adaptive decisions. Relationship modeling inspired by coreference and entity-based coherence allows the framework to interpret multiple events as components of a larger operational process (Baldwin & Morton, 1998; Barzilay & Lapata, 2005). Consequently, routing or prioritization decisions can consider historical relationships rather than relying solely on instantaneous metrics.

The fourth finding is that selective memory is essential for sustainable real-time intelligence. Long-context processing demonstrates the value of retaining broader context, while research concerning memory decay highlights the limitations associated with unrestricted retention (Beltagy, Peters, & Cohan, 2020; Berman, Jonides, & Lewis, 2009). The proposed separation of short-term, contextual, and persistent state provides a compromise between contextual richness and computational efficiency.

The fifth finding concerns resilience. Decentralized recovery can prevent an individual component failure from becoming a complete pipeline failure. However, resilience is not automatically guaranteed by agent multiplicity. Recovery decisions must preserve state consistency and prevent multiple agents from simultaneously modifying the same workload. Therefore, coordination remains a critical component of the framework.

Finally, the analysis indicates a fundamental trade-off between intelligence and overhead. More sophisticated agents can make more informed decisions, but their inference, communication, and state-management costs increase. The framework is therefore most appropriate when the benefits of adaptation exceed the computational cost of coordination. This observation reinforces the controllability principle associated with intelligent processing systems (Amplayo, Angelidis, & Lapata, 2021) and suggests that adaptive behavior should itself be dynamically controlled.

5. DISCUSSION

The proposed framework contributes to the theoretical integration of contextual AI and distributed stream processing. Existing literature demonstrates that intelligent processing benefits from controllability, contextual relationships, semantic coherence, and selective memory. The proposed architecture transfers these concepts into an event-streaming setting rather than treating streaming solely as a throughput optimization problem.

A key theoretical implication is that streaming intelligence should be state-aware rather than event-isolated. Conventional event processors frequently evaluate events individually or within short windows. However, real operational conditions often emerge from relationships across multiple events. Coreference-based summarization demonstrates the value of resolving relationships between information units (Baldwin & Morton, 1998), while lexical chains and entity-based coherence provide complementary perspectives on maintaining semantic continuity (Barzilay & Elhadad, 1997; Barzilay & Lapata, 2005). In a streaming architecture, these principles support the development of event graphs and contextual agent memory.

The framework also reveals a significant trade-off between centralized consistency and decentralized adaptability. Centralized coordination simplifies global state management but introduces bottlenecks and potential single points of failure. Decentralized agents improve autonomy and fault tolerance but increase the difficulty of maintaining globally consistent decisions. The architecture therefore requires coordination protocols that balance local responsiveness with global system objectives.

Another issue is the relationship between long context and computational efficiency. Longformer demonstrates that transformer-based processing can be adapted to longer contexts (Beltagy, Peters, & Cohan, 2020), but real-time systems cannot simply preserve unlimited event history. The proposed hierarchical memory model addresses this limitation by distinguishing immediately relevant information from compressed historical context. This is consistent with the broader problem of memory retention identified by Berman, Jonides, and Lewis (2009).

The framework further suggests that domain specialization is important for practical deployment. SciBERT demonstrates the value of domain-specific pretrained representations (Beltagy, Lo, & Cohan, 2019). Similarly, agents operating on technical, financial, industrial, or application-specific streams may require specialized semantic models. A general-purpose agent architecture may therefore provide the coordination infrastructure while allowing domain-specific models to operate inside individual agents.

The concept of controllability is equally significant. Amplayo, Angelidis, and Lapata (2021) demonstrate how model behavior can be conditioned according to desired aspects. In streaming, this translates into adaptive optimization objectives. A system processing safety-critical events may prioritize reliability, whereas a high-volume analytics platform may prioritize throughput. Consequently, there is no single universal definition of optimal stream processing.

Reddy et al. (2026) provide particularly relevant evidence for positioning multi-agent AI as an approach to optimized event streaming with improved scalability and resiliency. The proposed framework extends this direction by explicitly incorporating contextual memory, relationship modeling, adaptive objective selection, and hierarchical state management.

Nevertheless, several limitations remain. First, the framework has not been validated through controlled experiments, so quantitative claims concerning latency reduction, throughput improvement, or fault-recovery time require empirical verification. Second, multi-agent communication may itself become a scalability bottleneck. Third, inconsistent state synchronization can produce conflicting decisions. Fourth, sophisticated language models may impose substantial inference costs. Finally, autonomous adaptation introduces governance concerns because system behavior can change dynamically and may become difficult to audit.

Combinatorial optimization can provide an additional basis for managing scalability constraints in adaptive computing environments. Ramamurthy et al. (2026) present a framework for addressing scalability constraints through an LLM-based approach, highlighting the potential of intelligent optimization methods for supporting resource and workload management in complex computational systems.

Future validation should therefore evaluate the framework using controlled workloads involving burst traffic, heterogeneous event streams, node failures, changing semantic distributions, and resource constraints. Metrics should include end-to-end latency, throughput, recovery time, resource utilization, coordination overhead, state-consistency errors, and adaptation accuracy.

6. CONCLUSION

This paper proposed an adaptive multi-agent AI framework for real-time data streaming designed around scalability, resilience, contextual intelligence, and dynamic workload adaptation. The architecture replaces rigid processing pipelines with specialized agents capable of monitoring workload conditions, maintaining contextual state, allocating tasks, scaling resources, and recovering from failures.

The theoretical foundation combines controllable intelligent processing, relationship-aware representation, semantic coherence, long-context processing, and selective memory. These principles provide a conceptual basis for treating streaming events as connected computational information rather than isolated messages. The integration of these principles with multi-agent event-streaming research offers a pathway toward more autonomous and resilient streaming architectures.

The principal contribution is therefore architectural: scalability is achieved through selective agent replication and workload redistribution, while resilience is supported through decentralized recovery and state-aware task reassignment. At the same time, the analysis demonstrates that additional intelligence introduces communication, inference, synchronization, and governance costs.

Future research should transform the proposed conceptual model into a measurable implementation and evaluate it against static stream-processing architectures under controlled workload and failure conditions. Particular attention should be given to adaptive memory policies, agent coordination protocols, domain-specific models, and predictive resource allocation. Such empirical validation would determine whether the theoretical advantages of multi-agent adaptation translate into measurable improvements in real-time streaming performance.

REFERENCES

1. Amplayo, R. K., Angelidis, S., & Lapata, M. (2021). Aspect-Controllable Opinion Sum-marization. InProceedings of the 2021 Conference on Empirical Methods in NaturalLanguage Processing(pp. 6578–6593). Online and Punta Cana, Dominican Republic:Association for Computational Linguistics.

2. Baldwin, B., & Morton, T. S. (1998). Dynamic coreference-based summarization. InProceedings of the Third Conference on Empirical Methods for Natural Language Pro-cessing(pp. 1–6). Granada, Spain: Association for Computational Linguistics.

3. Barzilay, R., & Elhadad, M. (1997). Using Lexical Chains for Text Summarization. InIntelligent Scalable Text Summarization.

4. Barzilay, R., & Lapata, M. (2005). Modeling Local Coherence: An Entity-Based Approach.InProceedings of the 43rd Annual

Meeting of the Association for Computational Lin-guistics (ACL’05)(pp. 141–148). Michigan, USA: Association for ComputationalLinguistics.

5. Beltagy, I., Lo, K., & Cohan, A. (2019). SciBERT: A Pretrained Language Model for Sci-entific Text. InProceedings of the 2019 Conference on Empirical Methods in NaturalLanguage Processing and the 9th International Joint Conference on Natural LanguageProcessing (EMNLP-IJCNLP)(pp. 3615–3620). Hong Kong, China: Association forComputational Linguistics.

6. Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The Long-Document Trans-former.ArXiv preprint,abs/2004.05150.

7. Bengio, Y. (2017). The consciousness prior.ArXiv preprint,abs/1709.08568.

8. Berman, M. G., Jonides, J., & Lewis, R. L. (2009). In search of decay in verbal short-termmemory.Journal of Experimental Psychology: Learning, Memory, and Cognition,35(2), 317.

9. 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.

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. Geo Philip, Paulson, Artificial Intelligence and Machine Learning Applications in Project Schedule Forecasting: A Predictive Framework for Time-Control in Complex Building Projects (April 16, 2026). Available at SSRN: https://ssrn.com/abstract=6588119 or http://dx.doi.org/10.2139/ssrn.6588119

13. Vollem, S., Mulla, F. M., Shah, A. K., Kodela, S., Kaur, M., & Kumar, V. (2026, February). Multi-Model Time-Series Forecasting Framework for Stock Price Prediction Using Statistical and Deep Learning Techniques. In 2026 2nd International Conference on Big Data & Machine Learning (ICBDML) (pp. 1-6). IEEE.