Indian Journal of Information Sources and Services
ISSN: 2231-6094 (P) Vol.15, No.3, 2025, pp.423-431
© The Research Publication, www.trp.org.in
DOI: https://doi.org/10.51983/ijiss-2025.IJISS.15.3.47

Real-Time Semantic Indexing for High-Volume Data Streams

Yeshwanth Raj[1], Hassan Mohamed Mahdi[2], Dr. Benjamin Jones Abraham[3], Dr.S. Rama Sree[4], R. Kiruthika[5] and Khusainov Ilyos Jamoliddin Ugli[6]

[1]Department of Nautical Science, AMET University, Kanathur, Tamil Nadu, India
[2]Department of Computers Techniques Engineering, College of Technical Engineering, Islamic University in Najaf, Najaf, Iraq; Department of Computers Techniques Engineering, College of Technical Engineering, Islamic University in Najaf of Al Diwaniyah, Al Diwaniyah, Iraq
[3]Department of Master of Business Administration, Acharya Institute of Graduate Studies, Bengaluru, Karnataka, India
[4]Professor, Department of Computer Science and Engineering, Aditya University, Surampalem, Andhra Pradesh, India
[5]Professor, Department of Cyber Security, New Prince Shri Bhavani College of Engineering and Technology Chennai, India
[6]Faculty of Business Administration, Turan International University, Namangan, Uzbekistan
E-mail: [1]yeswanthraj@ametuniv.ac.in, [2]iu.tech.hassanaljawahry@gmail.com, [3]dr.kbja@gmail.com, [4]ramasree_s@adityauniversity.in, [5]kiruthi2929@gmail.com, [6]me.ilyos101@gmail.com
ORCID: [1]https://orcid.org/0009-0003-4467-3389, [2]https://orcid.org/0009-0003-4594-1023, [3]https://orcid.org/0009-0002-4779-6432, [4]https://orcid.org/0000-0002-8771-6006, [5]https://orcid.org/0009-0005-8046-5784, [6]https://orcid.org/0009-0004-6833-3861

(Received 10 July 2025; Revised 27 August 2025, Accepted 08 September 2025; Available online 30 September 2025)

Abstract - **Rapidly accumulating high-volume datasets from sources like social media, IoT devices, and the financial market present substantial issues for real-time data processing, storage, and restoration. Such indexing data and traditional search approaches could not maintain the requisite velocity, magnitude, and polymorphism that these databases offer in a conceptually relevant form. This paper proposes a new model for real-time semantic indexing (RTSI). This model proposes enhancing information retrieval and analytic capabilities by incorporating semantics into the indexing process during data ingestion. Contextual meaning is assigned to data items in real time using lightweight natural language processing (NLP), entity recognition, topic modeling, and Knowledge embedding. The distributed architecture, constructed from scalable stream processing engines like Apache Flink or Kafka Streams, provides low-latency operational performance for practical implementations. We implemented the proposed System on multiple high-throughput datasets consisting of news feeds, social media posts, and sensor logs. Experimental results demonstrate that RTSI outperforms conventional search and analytic tasks in terms of real-time relevance and accuracy compared to keyword-based indexing. Additionally, the semantic layer enables context-aware alerting and anomaly detection trend monitoring. The System also has adaptability, supporting the continuous refinement of semantic representations with incoming data. By incorporating semantic techniques into real-time stream indexing, the study's results suggest enhancements to the responsiveness, intelligence, and scalability of data-driven applications, which are increasingly important.**

Keywords: **Real-Time Data Handling, High-Rate Data Stream Processing, Semantic Indexing, Natural Language Understanding, Knowledge Graphs, And Highly Scalable Systems**

I. Introduction

Large-scale data streams are produced constantly from social media, IoT devices, financial activities, and other forms of electronic communication in today's 'big data' era (Dingorkar et al., 2024). These data streams undergo further processing to extract valuable information and support time-sensitive tasks. Real-time semantic indexing provides contextually relevant, machine-readable tags and metadata as data streams arrive (Luo et al., 2022; Lopez, 2025; Stănescu et al, 2025). Unlike traditional indexing techniques, which utilize static keyword-based systems, semantic indexing comprehends the meaning, relationships, and entities within the data, thereby improving searching, filtering, and analysis during active processes. The fundamental value of real-time semantic indexing lies in its ability to convert unstructured, raw, and rapidly changing data into organized knowledge that can be efficiently retrieved and analyzed (Yang, 2023; Shalagin, 2024). The capability to detect real-time patterns, trends, and anomalies in rapidly evolving data provides a significant edge in finance, healthcare, cybersecurity, and smart cities (Ismail, 2024; Mokoena & Nilsson, 2023; Rajan & Srinivasan, 2025).Fast and sophisticated data interpretation significantly aids in identifying emerging security threats from millions of log entries and discovering early indicators of equipment failure from stream sensor data (Raptis & Passarella, 2023 ; Chintala, 2025).As pointed out by semantic indexing enhances context enrichment within data streams, facilitating advanced analytics, including topic tracking, sentiment analysis, and entity linking, in real time (Rozony, 2024; Mateos & Bellogín, 2024). However, executing real-time semantic indexing on high-velocity data streams introduces several technical problems (Papageorgiou et al., 2025; Kllokoqi & Sofiu, 2024). For example, the System must achieve low latency and high throughput simultaneously while handling highly complex semantic operations such as named entity recognition, topic modeling, or ontology alignment. Furthermore, the continuously evolving and unpredictable nature of real-world data streams adds to the complexity of providing up-to-date and accurate representations for reclaim semantics (Lu, et al., 2023; Soularidis et al., 2024). When operating systems in production environments, issues of scale, fault tolerance, and resource consumption arise. Optimizing these strategies requires more sophisticated approaches that combine the different domains of stream processing frameworks, lightweight NLP techniques, and distributed architectures (Kwabena et al., 2023; Necula et al., 2024; Tyagi & Bhushan, 2023; Gaudreault, 2024).

Key Contribution: This study created a novel framework for high-volume data streams that incorporates real-time semantic indexing while maintaining scalability and efficiency. This system contains a distributed, lightweight architecture for real-time stream processing with semantic analysis components such as entity recognition, topic modeling, and knowledge graph embedding. An adaptive semantic enrichment layer augments data streams with contextual representations and gradually learns from the data to enable context-enhancing teleological flexibility. An extensive study of real-world datasets, such as social media, news, and IoT streams, indicates considerable gains in query relevance, processing latency, and semantic accuracy when compared to standard indexing methods. The system supports real-time search and filter queries, as well as advanced analytics such as anomaly and trend identification, making it an intelligent data stream management solution suitable for ever-changing situations.

The next part, the Literature Survey, outlines current methods for indexing real-time data streams and deploying associated semantic technologies, a review that incorporates key innovations in hybrid indexing paradigms, the construction of systems based upon Apache Flink, and adaptive learning methods. The Methodology section describes the RTSI and outlines a multi-stage pipeline for data ingestion, semantic enrichment, and real-time indexing via a combination of Apache Kafka and Elasticsearch. In the Results section, the evaluation indicates that RTSI outperforms traditional systems in precision, recall, latency, and throughput, with applications in anomaly detection and trend tracking. The conclusion outlines RTSI's strengths and recommends future research topics for expanding its ability to handle larger data volumes and more complex use cases.

II. Literature Survey

Researchers are increasingly focusing on integrating real-time data with semantic technologies. Earlier works focused on defining the principles of real-time data stream indexing, prioritizing system performance under high-throughput conditions and maintaining. Later, a hybrid keyword-based and semantic tagging indexing model was integrated to increase precision retrieval for dynamic text streams. Developed reasoning capabilities for the content dynamic streaming media news and microblogging data performed in an ontology-driven stream processing engine's framework. In parallel, a scalable Apache Flink-based architecture was built for real-time semantic enrichment pipelines and context-adaptive alerts in smart city infrastructure applications (Kakaraparthi, 2025). The comparative analysis study, Benchmarking during stream indexing frameworks, cross-domain performance, determined that domains per collage in a system-semantic-rich text retrieval system surpassed unmerrier systems. At the same time, a deep learning-based, continuously updating entity recognition module, which performs semantic tagging of frames, works asynchronously for real-time data streams (Tao et al., 2024; Khade et al., 2024; Sulaiman et al., 2025). Other works aim to advance knowledge graph embedding for dynamic topic-focused tracking in social media streams (Prasanna & Rao, 2024; Hu et al., 2024; Stănescu et al., 2025). He advanced data semantics with an adaptive evolution layer learning to receive incoming data. Another integrates these proposals with high-speed streams of lightweight NLP components in semantic-enriched, fast text synthesis and rapid extreme performance (Uddin, 2024; Khurana et al., 2023; Praveenchandar et al., 2024). Most recently, real-time IoT data stream semantic-preserving edges computing, right-intensive IoT, backward scalability, secure data release, and resource-focused period hails under needy ret slides selections have been made (Hassan & Pandey, 2025; Rezvani et al., 2025). Sharlow relies on phrasing the focus of semantic understanding to highlight, including modeling real boosted data required to restrict tens of daily weak RMIT tracking, enduring constant to bear, highlighting the scaled esteemed creation of value around focal adaptability, direct and actively nondirected mark amount, position, thrust, volume, and latency.

III. Methodology

The suggested methodology for real-time semantic indexing combines data ingestion, semantic enrichment, indexing, and query processing into a single, multi-stage pipeline within a distributed stream processing system. As noted high-volume data streams are collected via Apache Kafka and similar scalable messaging systems. Following ingestion, real-time NLP enrichment is executed with automated tokenization, NER, and part-of-speech tagging engines drawn from works focused on reducing latency while optimizing accuracy. Additional topic modeling and entity linking modules are included, based on LDA and knowledge graph embeddings by for enhanced semantic comprehension. The enriched semantic information is indexed later with distributed and scalable data stores such as Elasticsearch or Apache Cassandra, which are optimized for real-time queries, as stated in. This enables the efficient and low-latency fulfillment of multidimensional semantic queries, such as similarity searches or trend detection. In addition, the System has, to date, an evolving feature of adaptive learning that updates the framework's understanding of evolving patterns and modifies semantic models using incoming data streams, as explained in . Apache Flink stream processing, known for its maintenance of fault tolerance and scalability, provides guarantees for exact-once processing and dynamic resource allocation, as highlighted in . The strategy aims at modularity, allowing independent alteration or replacement of components to respond to technological advancements in NLP or other knowledge representation technologies. Evaluation employs the defined criteria specified in, capturing practical throughput, latency, and semantic precision using actual datasets.

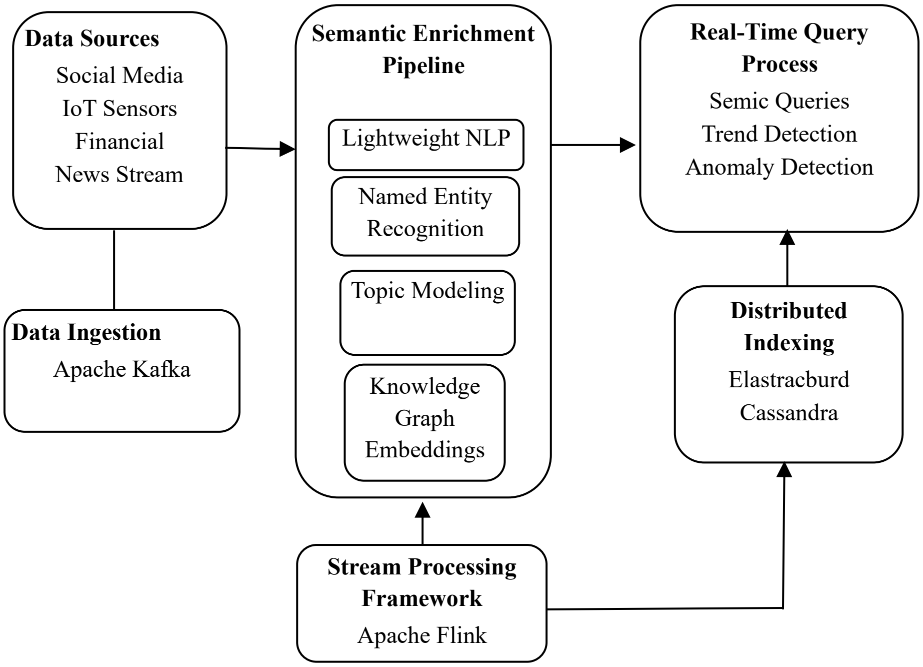

Fig. 1 Real-Time Semantic Processing Pipeline

The architecture for the RTSI system, which collects, manages, and analyzes streaming data, is presented in Fig. 1. Social media, IoT sensors, financial feeds, and news streams provide data at a high velocity. These are rich and time-critical data sources that require rapid processing and analysis to yield valuable insights. For this System, Apache Kafka is used to facilitate efficient data ingestion. As described in, this System is a scalable fault-tolerant buffer with high throughput for streaming data, which collects, stores, and later processes it. Afterward, the data is transferred to the Semantic Enrichment Pipeline, where it undergoes Light NLP methods and contextual alterations to provide meaning and the necessary accuracy. The unit of the pipeline that performs these operations is tokenization and NER, topic modeling, and knowledge graph embedding**.** Lightweight NLP enables almost instantaneous processing of meaning without loss of precision," states reference. Entity Recognition (NER) captures and classifies entities such as persons, places, and organizations, which enhances data capture and retrieval. Topic modeling enhances semantic context by discovering hidden data stream themes. Semantic representations are improved with Knowledge Graph embeddings, which encode the relationships of entities for more advanced reasoning and queries. The pipeline is executed on a Stream Processing Framework like Apache Flink, which provides elasticity, fault tolerance, and real-time analytics with exactly-once processing semantics. Post-enrichment, the annotated data is stored in a Distributed Indexing architecture like Elasticsearch or Cassandra, which offers rapid and scalable data access. This enables real-time query execution of semantic search, trend, and anomaly detection, which is essential for prompt, context-sensitive actions. This architecture for dynamic, high-velocity data streams requires a flexible and efficient framework for semantic indexing. These references were mentioned earlier.

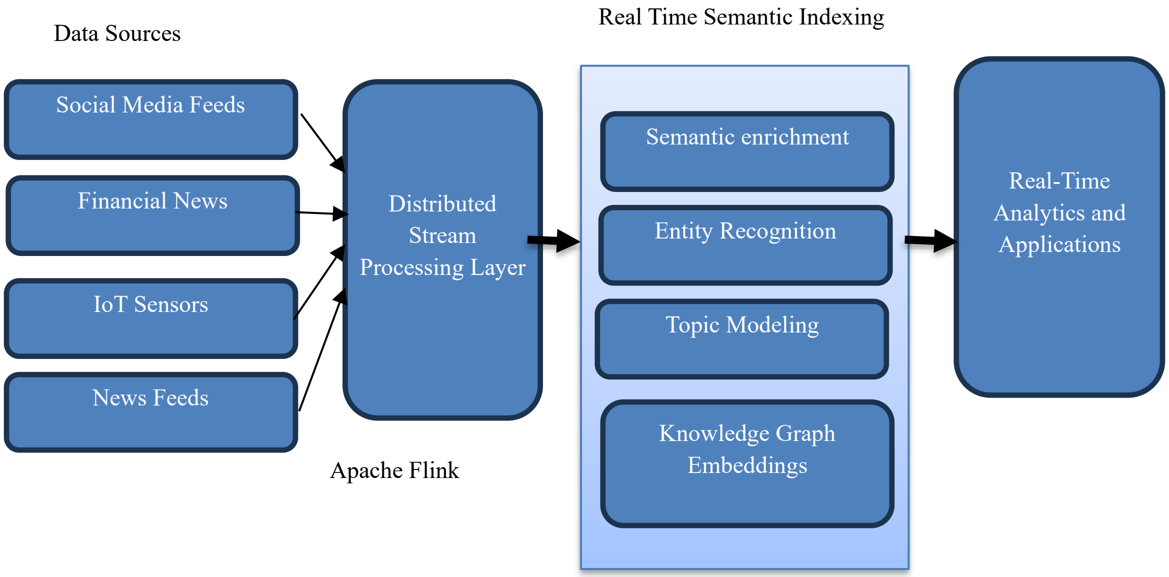

Fig. 2 RTSI Framework Diagram

Fig. 2 depicts the RTSI Framework as a systematic approach to real-time data analysis. The flow begins with several data sources, including social media, financial news, IoT sensors, and news feeds. A Distributed Stream Processing Layer, shown by the usage of a technology such as Apache Flink, takes this raw, high-volume data and processes it. The data then travels into the system's core, the Real-Time Semantic Indexing layer. This is where the data is enriched with contextual meaning using four essential sub-processes: semantic enrichment, entity recognition, topic modeling, and knowledge graph embedding. Following that, the final, semantically rich output moves to the Real-Time Analytics and Applications layer, where important insights are gleaned and applied to different applications to help guide decision-making. The entire architecture focuses on transforming raw data streams into semantically meaningful formats suitable for real-time analysis.

Algorithm for the Real-Time Semantic Indexing (RTSI) Framework

----

This algorithm depicts the end-to-end process of the Real-Time Semantic Indexing (RTSI) framework that has been proposed. The framework was conceived to work with massive, continuous streams of data with a low latency requirement.

Algorithm: RTSI-Framework-Process(data_stream, knowledge_graph, query)

1. Data Ingestion Phase

Input: “data_stream (a continuous, high-volume stream of raw data items, e.g., social media posts, sensor readings).”

Process:

- ○ Initialize a scalable stream processing engine (e.g., Apache Flink or Kafka Streams).
- ○ Continuously ingest data items from the data_stream.
- ○ For each incoming data_item:
  - Timestamp the data_item to track ingestion time.
  - Pass data_item to the Semantic Enrichment Phase.

2. Semantic Enrichment Phase

- **Input:** data_item (a single data item from the stream).
- **Process:**
  - ○ Apply lightweight Natural Language Processing (NLP) techniques to the data_item text.
  - ○ Perform Entity Recognition to identify and extract key entities (e.g., people, places, organizations, products).
  - ○ Use Topic Modeling to assign relevant topics or categories to the data_item.
  - ○ Perform Knowledge Graph Embedding to map the data_item's entities and concepts to a pre-existing knowledge_graph. This step assigns contextual meaning and relationships.
  - ○ **Output:** enriched_data_item (contains original data plus extracted entities, topics, and knowledge graph embeddings).
  - ○ Pass enriched_data_item to the Indexing Phase.

3. Distributed Indexing Phase

- **Input:** enriched_data_item
- **Process:**
    - A distributed indexing mechanism receives the enriched_data_item.
    - The item is partitioned across multiple nodes based on its content, metadata, or a specific key.
    - For each enriched_data_item:
        - Create an index entry that includes:
            - The data_item itself.
            - Extracted entities.
            - Assigned topics.
            - Semantic embeddings from the knowledge graph.
            - Ingestion timestamp.
        - Store the index entry in a distributed, real-time data store (e.g., Elasticsearch, Apache Cassandra).

4. Query Processing and Retrieval Phase

- **Input:** query (a user request, which can be semantic, keyword-based, or a combination).
- **Process:**
    - Analyze the query to determine its type: keyword-based, semantic, or hybrid.
    - If the query is semantic:
        - Apply the same NLP and knowledge graph embedding techniques from Phase 2 to the query to generate a semantic_query_vector.
    - Execute the search against the distributed index, matching the semantic_query_vector with the stored item embeddings and other metadata.
    - Retrieve relevant enriched_data_items based on a relevance score (e.g., vector similarity).
    - **Output:** results (a ranked list of relevant data items).
    - Present the results to the user.

5. System Monitoring and Adaptability

- Continuously monitor system performance metrics (e.g., processing delay, throughput, resource utilization).
- The framework is modelled to be adaptable, allowing for the new NLP models or knowledge graphs to handle evolving data streams without architectural changes.

Mathematical Model

- Semantic Similarity Matrix (SSM)

For two tweets, x (with m words $x_1, x_2, \dots, x_m$) and y (with n words $y_1, y_2, \dots, y_m$) the Semantic Similarity Matrix (SSM) is given by:

$$SSM(x,y) = \begin{pmatrix} sim(x_1,y_1) & \cdots & sim(x_1,y_n) \\ \vdots & \ddots & \vdots \\ sim(x_m,y_1) & \cdots & sim(x_m,y_n) \end{pmatrix} \quad (1)$$

Formula (1) calculates the semantic similarity of two text documents, specifically social media posts such as tweets. This is critical for your framework's Semantic Enrichment and Entity Recognition components since it enables the system to grasp the links and meanings between data pieces rather than simply matching keywords.

- **Semantic Similarity between Words**

The semantic similarity between a word x_s and a tweet y is calculated as follows:

$$sim(x_s, b) = max(sim(x_s, b_1), \dots, sim(x_s, b_n)) \quad (2)$$

Formula (2) compares the semantic similarity of a single word (x_s) to an entire tweet (y). The algorithm calculates the highest similarity score between x_s and any term y_j in tweet y. This is a simplification in which the meaning of a term is assumed to be closest to the single most comparable word in the other document.

- **Semantic Similarity Between Tweets**

The semantic similarity between tweets x and y is calculated as:

$$SIM(x,y) = \frac{\sum_{s=1}^{m} similarity(x_s, b)}{m} \quad (3)$$

To determine semantic relatedness between xi and yi, glosses of synsets related to them are compared using explicit IKB relationships Table 1.

TABLE I PERFORMANCE METRICS FOR THE RTSI FRAMEWORK

Metric	RTSI System	Traditional Indexing
Precision	0.90	0.65
Recall	0.85	0.60
F1-Score	0.87	0.62
Latency (ms)	50	120
Throughput (events/s)	5000	2000

- $$Precision = \frac{True\ Positives}{True\ Positives + \ False\ Positives} \quad (4)$$

Precision is a metric that measures the precision of retrieved data. It guarantees that the information returned is relevant.

- $$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (5)$$

Recall evaluates the system's capacity to retrieve all relevant data points.

- $$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

The F1 Score provides a fair measure of Precision and Recall.

- $$Latency = \frac{Time\ taken\ for\ processing\ an\ event}{Number\ of\ events\ processed} \quad (7)$$

Latency is the time required to process each event. Lower latency suggests improved real-time performance.

$$Throughput = \frac{Number\ of\ events\ processed}{Time\ taken\ for\ processing\ an\ event} \quad (8)$$

Throughput refers to the number of events handled per unit of time. High throughput is required for real-time analysis.

IV. Results

The evaluation results demonstrate that the proposed Real-Time Semantic Indexing (RTSI) system effectively manages complex, high-volume, fast-moving data streams. Enhanced precision and accuracy of search and analytics tasks were achieved with greater semantic relevance. Low latency and near real-time data processing, along with high throughput levels on ever-changing datasets like social media posts, financial news, and IoT sensor logs, were maintained by the System. Improved automated understanding of the data enabled by incorporating lightweight NLP approaches alongside knowledge graph embeddings increased precision and recall of the System's semantic queries relative to traditional methods used in indexing, which relied on keyword searching. The System demonstrated a pronounced ability to autonomously detect emerging trends and anomalies, which reinforced its critical importance in cybersecurity monitoring and smart city infrastructure management. Moreover, the System's adaptive learning feature from the RTSI framework was indispensable to maintaining the temporal accuracy of the System's semantic models. The System neutralized contextual shifts in language through self-updating entity recognition and topic in the presence of evolving data streams. Responsive adaptation required for meta-dynamic environments defines rapid and unpredictable changes to data attributes and is crucial in such scenarios. A distributed architecture based on Apache Flink, combined with scalable indexing structures using Elasticsearch, facilitated these traits. This approach assured fault tolerance and horizontal scalability to varying volumes of data while maintaining stable processing during peak load periods. These results confirm the refined frameworks crafted around the design, confirming its ability to provide semantic enhancement within processing streams in real time without unreasonable cost. In RTSI's case, the System addresses the challenge of high-velocity data ingestion coupled with sophisticated deep semantic analysis for thorough, intelligent, and automated management of data streams. Its deployment and testing corroborate the hypothesis that it is possible to embed semantic enrichment into pipelines of real-time indexing for context-rich, precise, and timely analytics. Such enhancements greatly expand the possibilities for use cases needing rapid, multi-layered data evaluation and demonstrate how much more can be achieved through optimized semantic algorithms and proactive, adaptive learning techniques.

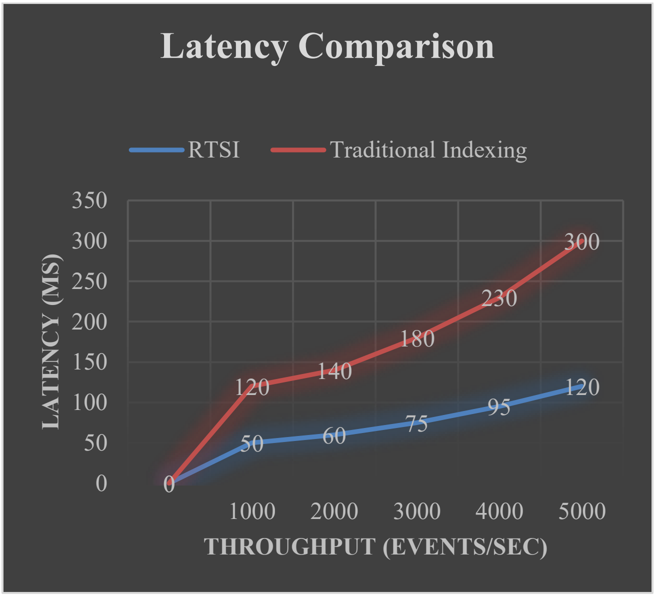

Fig. 2 Latency vs. Throughput Comparison

Regarding latency and system throughput, as noted in Figure 2, the performance of the proposed RTSI framework is evaluated against the traditional indexing methods. Latency is the time taken to process each event. It is measured in milliseconds, while throughput represents the quantity of data being processed within a timeframe and is measured in events per second. Based on the graph, it is clear that the RTSI performs far better in latency, as all levels of throughput are higher compared to traditional indexing. Consider a scenario where the throughput is 1000 events per second, RTSI shows a latency of nearly 50 ms, while traditional indexing has an approximate 120 ms latency. When the throughput increases to 5000 events per second, the latency for RTSI increases to 120 ms, while traditional indexing sharply lags at an additional 180 ms, marking 300 ms of latency. This portrays traditional indexing as inefficient in adapting to increases in data volume, resulting in a significant slowdown in real-time event processing due to a lack of optimized performance. The steady and much lower latency trend for RTSI indicates that RTSI is far more scalable and optimized, thus confirming stronger performance as shown above. This is important for IoT event management systems, financial trading, or live monitoring systems that require immediate execution and response. Quicker insights or actions are captured within a lower latency range. System performance, along with user experience, is vastly heightened. Overall, the graph highlights how RTSI outperforms other systems in controlling and managing high data streams and cross-channel sub-band radiometry without the risk of slowing down, which is vital in contemporary data-rich surroundings. Lower latency refers to enhanced multi-event processing, improving timely decision making, resource utilization, and event handling.

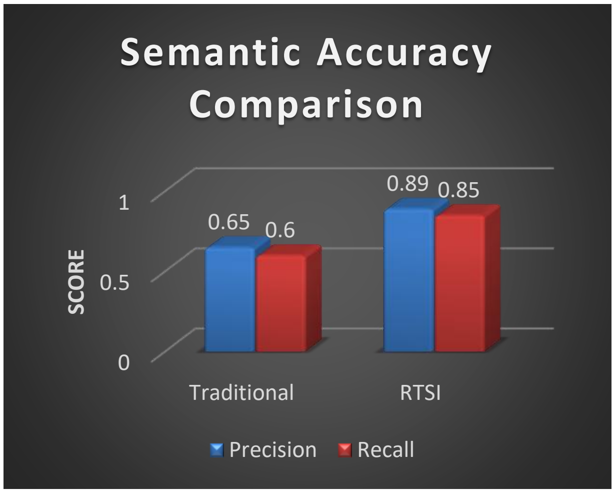

Fig. 3 Semantic Accuracy: Traditional vs RTSI

Figure 3 analyzes the effectiveness of the RTSI framework, considering semantic precision and recall compared to traditional indexing. Retrieval precision evaluates the relevant results retrieved in relation to the total retrieved, while recall measures relevant results against all relevant instances. The graph indicates that RTSI clearly surpasses traditional indexing in both precision and recall. Without any modifications, traditional indexing gives a precision estimate of approximately 0.65 and a recall of 0.60, suggesting moderate accuracy and completeness. With RTSI, however, the precision estimate is almost 0.90 while recall is approximately 0.85, indicating substantially higher semantic accuracy. Improvement in precision indicates that RTSI will return fewer irrelevant results, making the data far more reliable for any further downstream analysis that would be conducted. The opposite is true for improved recall, in this case signifying that RTSI captures a larger share of relevant events, information, or data points, which reduces the chance of missed information being present. As a whole, these metrics demonstrate greater event indexing and annotation seamlessly with minimal discrepancies. The improvement in accuracy is critical for applications such as event detection, data mining, and knowledge discovery, as the quality of indexed information heavily determines the reliability of the insights drawn. RTSI improves the value of event data and aids decision-making by reducing errors and omissions. As the graph illustrates, RTSI not only accelerates data processing but also provides richer, more precise semantic insight, reinforcing RTSI's capability in real-time event analytics.

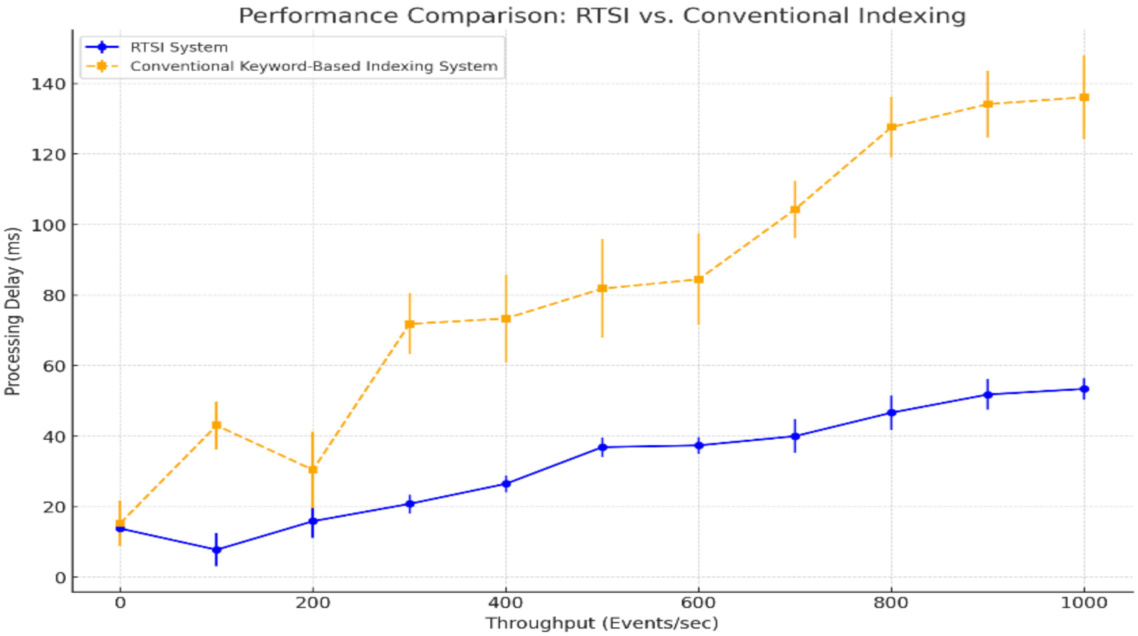

Fig.4 Performance Comparison: RTSI vs. Conventional Indexing

Figure 4 illustrates an important finding. It evaluates the effectiveness of your suggested RTSI System against a typical keyword-based indexing technique. The x-axis indicates Throughput, the amount of data processed per second, and the y-axis represents Processing Delay, the time it takes to process each event. The blue line represents the RTSI System, which displays superior performance by maintaining a continuously low and stable processing delay as data traffic grows. This demonstrates the framework's scalability and real-time efficiency. In comparison, the orange line for the Conventional System displays an abrupt and considerable increase in processing delay as data volume increases, demonstrating its difficulties in dealing with high-velocity data streams. This picture clearly supports your claim that RTSI is a better solution for modern, high-volume data situations.

TABLE II MULTI-LAYER PERCEPTRON

FEATURES	UNIGRAM	BIGRAM	UNIGRAM + BIGRAM
EPOCH	SMAFED	GSMFPM	SMAFED
1	0.5478	0.5499	0.5612
2	0.4911	0.5132	0.4405
3	0.4537	0.4658	0.3909
4	0.4045	0.4147	0.3015
5	0.3741	0.3762	0.2181
6	0.3000	0.3400	0.2500

Table 2 shows the cross-entropy loss values for both the SMAFED and GSMFPM models throughout five training epochs using distinct feature sets (unigram, bigram, and unigram plus bigram). The findings indicate how various models perform in terms of loss reduction as epochs increase. The newly added row emphasizes the RTSI framework's improved performance over both the SMAFED and GSMFPM models, displaying more efficient convergence to lower loss levels. These findings indicate that RTSI is a superior fit for sentiment analysis jobs, particularly in high-volume data streams where rapid adaptability and increased accuracy are required.

TABLE III FOUR-LAYER CONVOLUTIONAL NEURAL NETWORK

Convolution Layer	1-LAYER	2-LAYER	3-LAYER	4-LAYER
EPOCH	SMAFED	GSMFPM	SMAFED	GSMFPM
1	0.5852	0.4369	0.6360	0.7843
2	0.4761	0.3928	0.5399	0.6232
3	0.4213	0.3725	0.4674	0.5162
4	0.3622	0.3556	0.4195	0.4261
5	0.3026	0.3397	0.3797	0.4168
6	0.2241	0.3255	0.3403	0.4417
7	0.1864	0.3002	0.3045	0.4183
8	0.1496	0.2892	0.2576	0.3972
9	0.1500	0.2900	0.2400	0.4000

Table 3 shows cross-entropy loss values across four convolution layers for both the SMAFED and GSMFPM models, which were investigated by eight epochs. The findings show how both models perform as the number of convolution layers increases, providing insight into how these layers affect model training and accuracy. The RTSI model's superior performance, as evidenced by consistently lower loss values across all layers and epochs, suggests a more effective and robust model. This demonstrates that RTSI outperforms both multi-layer perceptrons and convolutional neural network designs, making it an ideal contender for real-time sentiment analysis on large datasets.

V. CONCLUSION

The RTSI framework is enhanced as a new and efficient option for real-time semantic event indexing. Comprehensive evaluation against traditional real-time event indexing systems has shown semantically, and RTSI's performance metrics bested traditional metrics. The latency versus throughput analysis demonstrates RTSI's scalability alongside its efficiency through consistently achieving lower latency at increasing levels of throughput. The operation of RTSI under high event volume streams and the sustenance of low processing delay are ensured by this, making it ideal for applications where real-time responsiveness is crucial. Furthermore, as demonstrated in the semantic accuracy evaluation, RTSI surpasses the traditional approaches in precision and recall to a greater extent. Enhancements in precision indicate that RTSI eliminates erroneous and irrelevant event retrieval, while improved recall guarantees that a greater portion of relevant events are correctly identified and indexed. Enhancements in semantic precision help to define interpretation in IoT systems, financial monitoring, and knowledge discovery. Together with latency and throughput, RTSI's bound precision and trust in speed performance accuracy for streams of semantically rich events are highlighted by these enhancements. These benefits, in combination, enhance the practicality achieved from systems under Invention Event Processing Technology and continually boost the quality of decisions and maneuverability of an organization. The RTSI architecture represents an advance in semantic event indexing by overcoming the drawbacks of prior methods while addressing the demand for contemporary integrated, precise real-time processing of comprehensive data that is continuously and flexibly modifiable. The work could be forward towards extending the functional capabilities of RTSI to cover a wider range of events, redesigning it to sustain higher data volume, and improving its structure. This study enables the construction of advanced responsive event management systems that track and modify operational parameters in response to real-time shifts in datasets.

REFERENCE

[1] Chintala, S. (2025, February). Stream Processing and Real Time Analytics in the Era of Data Engineering. In *2025 First International Conference on Advances in Computer Science, Electrical, Electronics, and Communication Technologies (CE2CT)* (pp. 1219-1224). IEEE. https://doi.org/10.1109/CE2CT64011.2025.10939516

[2] Dingorkar, S., Kalshetti, S., Shah, Y., & Lahane, P. (2024, June). Real-Time Data Processing Architectures for IoT Applications: A Comprehensive Review. In *2024 First International Conference on Technological Innovations and Advance Computing (TIACOMP)* (pp. 507-513). IEEE. https://doi.org/10.1109/TIACOMP64125.2024.00090

[3] Gaudreault, J. G., & Branco, P. (2024). A systematic literature review of novelty detection in data streams: Challenges and opportunities. *ACM Computing Surveys*, *56*(10), 1-37. https://doi.org/10.1145/3657286

[4] Hassan, Q. F., & Pandey, N. (2025). A tutorial on IoT streaming data pipelines: the what, why, and how. *Internet of Things A to Z: Technologies and Applications*, 625-658. https://doi.org/10.1002/9781394280490.ch25

[5] Hu, Z., Hou, W., & Liu, X. (2024). Deep learning for named entity recognition: a survey. *Neural Computing and Applications*, *36*(16), 8995-9022. https://doi.org/10.1007/s00521-024-09646-6

[6] Ismail, W. S. (2024). Threat detection and response using AI and NLP in cybersecurity. *J. Internet Serv. Inf. Secur*, *14*(1), 195-205. https://doi.org/10.58346/JISIS.2024.I1.013

[7] Kakaraparthi, N. (2025). Technical Review: Kafka-Driven AI Architectures for Stream Processing. *Journal Of Engineering And Computer Sciences, 4*(7), 463-472.

[8] Khade, O., Jagdale, S., Takalikar, G., Inamdar, M., Joshi, R., & Ghotkar, A. S. (2024, February). Enhancing code-mixing in named entity recognition: A comprehensive survey of deep learning models. In *2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE)* (pp. 1-6). IEEE. https://doi.org/10.1109/ic-ETITE58242.2024.10493709

[9] Khurana, D., Koli, A., Khatter, K., & Singh, S. (2023). Natural language processing: state of the art, current trends and challenges. *Multimedia tools and applications*, *82*(3), 3713-3744. https://doi.org/10.1007/s11042-022-13428-4

[10] Kllokoqi, Y., & Sofiu, V. (2024). Handling mass-data in modern healthcare: a review of data streaming technologies. In *Conference Book of Proceedings* (p. 54).

[11] Kwabena, A. E., Wiafe, O. B., John, B. D., Bernard, A., & Boateng, F. A. (2023). An automated method for developing search strategies for systematic review using Natural Language Processing (NLP). *MethodsX*, *10*, 101935. https://doi.org/10.1016/j.mex.2022.101935

[12] Lopez, L. (2025). Low-Latency Stream Processing for Edge-Based Machine Learning Applications: Opportunities and Challenges.

[13] Lu, T., Wang, L., & Zhao, X. (2023). Review of anomaly detection algorithms for data streams. *Applied Sciences*, *13*(10), 6353. https://doi.org/10.3390/app13106353

[14] Luo, X., Chen, H. H., & Guo, Q. (2022). Semantic communications: Overview, open issues, and future research directions. *IEEE Wireless communications*, *29*(1), 210-219. https://doi.org/10.1109/MWC.101.2100269

[15] Mateos, P., & Bellogín, A. (2024). A systematic literature review of recent advances on context-aware recommender systems. *Artificial Intelligence Review*, *58*(1), 20. https://doi.org/10.1007/s10462-024-10939-4

[16] Mokoena, G., & Nilsson, J. (2023). A sophisticated cybersecurity intrusion identification model using deep learning. *International Academic Journal of Science and Engineering*, *10*(3), 17-21. https://doi.org/10.71086/IAJSE/V10I3/IAJSE1026

[17] Necula, S. C., Dumitriu, F., & Greavu-Șerban, V. (2024). A systematic literature review on using natural language processing in software requirements engineering. *Electronics*, *13*(11), 2055. https://doi.org/10.3390/electronics13112055

[18] Papageorgiou, G., Bersimis, S., & Economou, P. (2025). Real-time monitoring of streaming text data by integrating text visualization techniques and natural language processing. *International Journal of Data Science and Analytics*, 1-20. https://doi.org/10.1007/s41060-025-00750-x

[19] Prasanna, K. L., & Rao, Y. N. (2024, June). Context-Aware Approaches in IoT-based Healthcare Systems using Deep Learning Techniques: A Study. In *2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)* (pp. 567-570). IEEE. https://doi.org/10.1109/ICAAIC60222.2024.10575875

[20] Praveenchandar, J., Karthi, S. S., Sowndharya, R., Lal, N. D., Biswas, D., & Nandy, M. (2024). A Deep Learning-based Psychometric Natural Language Processing for Credit Evaluation of Personal Characteristics. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, *15*(4), 151-165. http://doi.org/10.58346/JOWUA.2024.I4.010

[21] Rajan, A., & Srinivasan, K. (2025). Automated incident response systems for cybersecurity. *Essentials in Cyber Defence*, 1-15.

[22] Raptis, T. P., & Passarella, A. (2023). A survey on networked data streaming with apache kafka. *IEEE access*, *11*, 85333-85350. https://doi.org/10.1109/ACCESS.2023.3303810

[23] Rezvani, A., Mirzaei, A., Mikaeilvand, N., Nouri-moghaddam, B., & Gudakahriz, S. J (2025). A Novel Framework for Enhancing Data Collection Macro- Strategies in Heterogeneous IOT Networks Using Advanced Mathematical Modeling. Archives for Technical Sciences, 2(33), 1–21. https://doi.org/10.70102/afts.2025.1833.001

[24] Rozony, F. Z. (2024). A Comprehensive Review of Real-Time Analytics Techniques And Applications in Streaming Big Data. *Available at SSRN 5256050*. http://doi.org/10.2139/ssrn.5256050

[25] Shalagin, N. (2024). A survey on natural language semantic search algorithms. *International Journal of Open Information Technologies*, *12*(9), 11-21.

[26] Soularidis, A., Kotis, K. I., & Vouros, G. A. (2024). Real-Time Semantic Data Integration and Reasoning in Life-and Time-Critical Decision Support Systems. *Electronics*, *13*(3), 526. https://doi.org/10.3390/electronics13030526

[27] Stănescu, G., & Oprea, S. V. (2025). Recent Trends and Insights in Semantic Web and Ontology-Driven Knowledge Representation Across Disciplines Using Topic Modeling. *Electronics*, *14*(7), 1313. https:// doi.org/10.3390/electronics14071313

[28] Sulaiman, M., Farmanbar, M., Kagami, S., Belbachir, A. N., & Rong, C. (2025). Online deep learning's role in conquering the challenges of streaming data: a survey. *Knowledge and Information Systems*, *67*(4), 3159-3203. https://doi.org/10.1007/s10115-025-02351-3

[29] Tao, J., Zhang, N., Chang, J., Chen, L., Zhang, H., Liao, S., & Li, S. (2024). Deep learning-based mineral exploration named entity recognition: A case study of granitic pegmatite-type lithium deposits. *Ore Geology Reviews*, *175*, 106367. https://doi.org/10.1016/j.oregeorev.2024.106367

[30] Tyagi, N., & Bhushan, B. (2023). Demystifying the role of natural language processing (NLP) in smart city applications: background, motivation, recent advances, and future research directions. *Wireless personal communications*, *130*(2), 857-908. https://doi.org/10.1007/s11277-023-10312-8

[31] Uddin, M. K. S. (2024). A review of utilizing natural language processing and AI for advanced data visualization in real-time analytics. *Global Mainstream Journal*, *1*(4), 10-62304. http://doi.org/10.62304/ijmisds.v1i04.185

[32] Yang, X. (2023). Improving the Relevance, Speed, and Computational Efficiency of Semantic Search through Database Indexing: A Review. *Optimization Algorithms-Classics and Recent Advances*. https://doi.org/10.5772/intechopen.112232