International Journal of Enhanced Research in Science, Technology & Engineering
ISSN: 2319-7463, Vol. 9 Issue 11, November-2020, Impact Factor: 6.754

Real-Time Data Processing in Data Warehousing: Integrating SQL Warehouses with In-Memory Analytics

Rajesh Reddy Devireddy

University of Central Missouri, USA

ABSTRACT

This research focuses on using in-memory analytics with SQL data warehouses for fast analysis of data. The paper critiques batch-oriented tools for their limitations and instead explores modern frameworks that enable uninterrupted data acquisition, rapid transformation, and immediate query responses. Architectural design, system coordination, and performance optimization are emphasized as key priorities.Through secondary qualitative analysis, the study uncovers aspects of system adaptability, cloud scalability, and multi-repository integration.It explains in detail the use of real-time platforms that improve enterprise analytics. The key study results help data engineering teams design flexible, growing and standardised real-time warehouse systems for companies that experience constant changes.

Keywords: Real-Time Data Processing, SQL Data Warehouse, In-Memory Analytics, Streaming Data, ETL Optimization, Cloud-Native Platforms, Apache Spark, Apache Flink, Snowflake, Databricks, Big Query, Data Synchronization.

INTRODUCTION

Real-time understandings reshape enterprise data infrastructure. Batch-oriented ETL is used in traditional SQL-based data warehouses, leading to delays in getting data and making it hard to act fast. Processing information just once or in regular intervals is no longer effective in today's fast-changing business settings. This research looks at the move to architectures that combine SQL warehouses with in-memory processing for easy access to data and speedy analytics. Industry reports, case studies and technical documentation have been used for the secondary qualitative analysis conducted in the study. The system specialises in collecting streams, instant data processing and fast queries. This is made possible through platforms like Apache Spark, Apache Flink, Databricks, Snowflake, and Big Query.These platforms allow data to be used and analysed right away thanks to using memory-optimised engines.

Systems and methods are examined to improve the structure and speed up the entire process. Examples of usage and scenarios that explain the working principles of blockchain used and made better for specific tasks. The analysis makes the challenges of mixing SQL queries with streaming data very clear. The research illustrates the evolution of enterprise data warehouses into actionable, real-time decision-support systems. The new system depends on nonstop processing, built-in analytics and cloud expansion, making it possible for businesses to operate much faster and more flexibly.

Aim

The research aim is to analyze the integration of SQL-based data warehouses with in-memory analytics technologies for enabling real-time data processing and decision support in enterprise environments.

Objective

- To identify the limitations of traditional batch-based ETL workflows in dynamic data environments.
- To examine the capabilities of in-memory and cloud-native platforms for real-time data processing.
- To evaluate integration frameworks that combine SQL querying with streaming analytics.
- To assess performance, latency, and synchronization challenges in real-time data warehousing.

Research Question

1. What are the limitations of batch-based ETL workflows in supporting real-time analytics?
2. How do in-memory and cloud-native tools enable continuous data ingestion and transformation?

3. What integration models exist for combining SQL-based querying with real-time data streams?
4. What performance and synchronization issues arise in real-time data warehousing systems?

Research Rationale

Faster business decisions require real-time data in today's digital business environment. Most traditional data warehouses depend on batch-based ETL that results in delayed reporting and outdated information. These systems become unreliablewhen faced with continuously incoming data. Processing data instantly helps to compete, act fast and deal with changes in the business environment. In-memory analytics platforms and cloud-native solutions make it possible to work with data quickly, add more power and features whenever needed and connect with other systems [1]. It can use tools such as Apache Spark, Apache Flink, Snowflake and Databricks to handle steady streams, make fast changes and see the results in a proper way. They make data warehousing better by removing hurdles to performance and allowing for steady analysis.

The research looks at the movement from old ETL systems to having a real-time architecture in place. It relies on secondary analysis of technical reports, case studies and industry documents to find out whether the working principles work well or not in cybersecurity [2]. During the analysis, effort is placed on making software work faster, choosing an architectural style and seamlessly connecting pieces. Real-time models are used by companies to provide faster results, boost user satisfaction and coordinate operations more smoothly. Grasping synchronization techniques, latency and methods of deploying systems improves the system's design. The findings of this study help to show SQL-based warehousing that can keep up with today's real-time data processing. The study also provides actionable insights for data engineering and business intelligence teams.

LITERATURE REVIEW

Evolution of Traditional Data Warehousing

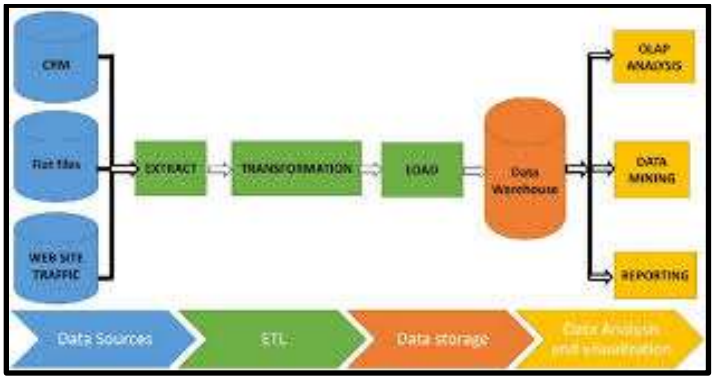

Figure 1: Traditional Data Warehousing

Data warehousing started with the focus on saving, handling and reviewing data in a structured way from transaction systems. They carried out the processing of data through batch-based ETL pipelines at planned moments. The model made it possible to produce reports on history, dashboard information and static data summaries [3]. The focus remained on consistency, reliability, and centralized storage. There are no significant delays in these environments since demand is easy to predict. Such systems have to run for a long time, cannot grow as quickly as wanted and sometimes deliver answers after a delay. Organizations tolerated time gaps between data collection and reporting. The traditional model struggled with several problems in its operation because business needs changed.

Limitations of Batch-Oriented ETL Workflows

SQL-based ETL systems work by first collecting data, processing it and finally putting it into SQL warehouses at set times [4]. These systems create blockages, especially during peak data loads. As long as data is not available in a timely manner, responses are slower, and the business becomes less flexible. Alterations to data sources or formats often cause problems during the workflow that resulting in more work. It is vital to adapt quickly and make static pipelines in the place there is a high volume of work. Real-time decisions demand low-latency processing and frequent data refreshes. Scheduling jobs manually, dealing with frequent errors, and following set configurations decrease the performance of old ETL systems. Supporting streaming data requires extra effort from the system. This can slow down performance.

Rise of In-Memory Data Processing

In-memory technologies emerged to accelerate data access and computation. Memory allows information in the system to be easily queried, changed and combined, making it quicker. Apache Spark and Apache Flink use distributed computing to run various operations in multiple parts at the same time. Waiting for I/O is reduced, and the transformation is very fast because data travels in memory-based pipelines [5]. It takes only seconds to run analytical queries, allowing users to explore, observe trends and find unusual situations. Large workloads are managed at high speed and consistency with the help of dynamic resource allocation in memory. These systems, enterprises can be both agile and scalable to fit the expectations of today.

Role of Cloud-Native Platforms

Businesses get increased memory and can manage resources more easily with the help of cloud-native platforms. Both Snowflake, Big Query and Databricks have auto-scaling, isolated resources and integrated streaming. Separating compute from storage in these platforms helps users optimize their resources and expenses. These frameworks rely on streaming data importing, event-triggered processing and analytics that happen instantly. According to [6], using cloud-native tools helps flawless integration with IoT devices, transactional systems and different APIs from third-party companies. The dashboards, alerts and workflows constantly work with the latest sets of information. Cloud ecosystems support agility, fault tolerance, and global availability.

Architectural Models for Real-Time Integration

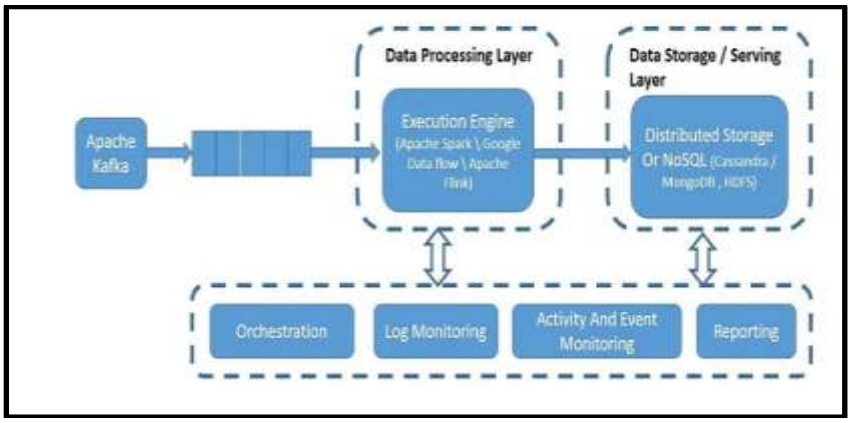

Figure 2: Kappa Architecture

The Architectural foundation in modern data architectures supports controlling data from batch and streaming pipelines together. Data processing in the Lambda architecture happens in batches and as real-time events and the results are merged into one output. The Kappa architecture reduces the model by making streaming the main engine for working with both historical and real-time data [7]. Message systems such as Kafka bring in data for processing in engines, and the data is then stored in analytical layers. Using SQL, users can query data in real-time and the past through these systems' interfaces. Such models are perfect for continuous analytics because they enable flexible, scalable and low-latency architecture.

Synchronization Between SQL and Streaming Engines

Real-time systems maintain structured queries while continuously adapting to changing data. The accuracy of Retorted in volatile settings is guaranteed by its use of consistency models, event time service and mature state management system. According to [8], using watermarking, windowing and checkpointing makes sure streamed results are in line with the SQL query results. Applications that must perform quickly require deterministic behavior in each processing node. Stream processing engines are equipped to handle events received in a different order, repeated messages and changes in the schema. It is essential to align data synchronization across multiple repositories for analysis to be right and data to be kept safe. Integration layers convert data, guarantee its delivery and help between different applications and systems.

Performance Optimization Strategies

Low-latency analytics depend on optimized processing pipelines. Techniques include partitioning, caching, and vectorized execution [9]. Query planner'spriorities resource-efficient operations, minimizing computation time. Data from the database is scanned less often, and the system is quicker with indexing, pruning and pushdown filters. Caching results in memory speeds up the application, especially while many users are working simultaneously. Change in workload does not force the system to slow down, using load balancing, backpressure and scaling. Real-time systems have to deal with bursts

in data, shifts in data structures and changing user commands in an automatic way. A good resource management and adaptive execution framework maintains the same performance when facing various workloads.

Deployment Considerations and Use Cases

Real-time systems are put in place after considering the amount of data, its rate of change, the system requirements and the users' requirements. Some uses for AI are spotting fraud, boosting supply chain processes, tailoring services based on real-time data and predicting equipment or machine breakdowns [10]. Being able to access data right away helps with maintaining constant monitoring, fast alerting and effective data analysis. As deployments are successful, they line up the infrastructure with the needs, picking the right mix of engines, tools and processes. Security, handling of data and compliance always be part of real-time architecture. Encryption and audit logging are all in place to keep private information private, and role-based access. Most deployment blueprints contain CI/CD, version control and automated testing to help ensure delivery is consistent.

Business Impact and Operational Agility

Real-time data warehousing transforms static operations into agile environments. Those involved in businesses are able to review dashboards in real time, immediately receive feedback and change their approaches accordingly. Various teams monitor performance in real time, notice any issues and apply automated response processes. Campaigns are targeted at individuals using recent and current customer behavior. According to [11], supply chains change the delivery routes or inventories to keep going in the case of problems. These capabilities improve decision speed, accuracy, and competitiveness. Continuous analytics drives innovation, customer satisfaction, and operational efficiency. Having information in real-time makes it easy to plan events and handle risks in advance across areas of the organization.

Literature Gap

The research has been done on real-time analytics, in-memory processing and SQL-based data warehousing. Much of the focus is placed on reviewing separate metrics, current system setups, or the platforms compared with one another. There are few studies on unifying traditional SQL databases with streaming engines in-memory [11]. Minor engagement is given to using blockchain in practice, the best ways to synchronize data and cases in that place blockchain can be applied. Business impact analysis does not include a very detailed technical design. Coming together, cloud-native tools, useful querying and instant readiness are still not fully explored. There is no detailed guide explaining real-time data processing that changes in SQL-based analytics systems in real life. As a result, data engineers and enterprise leaders do not have enough practical advice.

METHODOLOGY

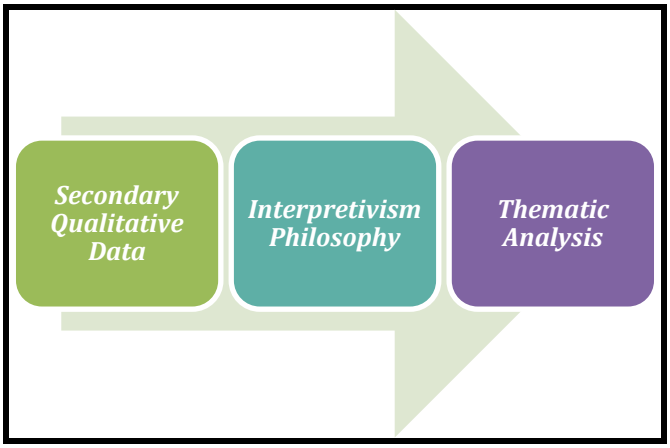

Fig. 2: Methodology

The research utilised a ***qualitative methodology*** by examining and organising existing research findings. This study adopts an interpretivist philosophy, which aligns with its aim to explore enterprise real-time data warehousing behaviors and system interactions. Using this philosophy, it is easy to analyse and find patterns and significance in current narratives and executed uses [12]. This analysis is constructed by using an ***inductive research approach*** that helps build themes from the qualitative data from various journals news, or other details [13]. The research aims to analyze industry developments, operational patterns, and technological shifts without relying on predefined hypotheses.

Part of data collection is to gather documents such as white papers, documentation from vendors, user manuals, case studies from the sector and technical articles reviewed by scholars. Factors considered for selection are related to SQL data warehousing, systems that process data in real time, in-memory analysis and platforms native to the cloud. Most important are books that concentrate on providing descriptions of issues that occur during implementation, details of the system design and techniques for boosting performance.

Data analysis employs ***thematic coding***. All the sources that have been chosen are inspected for topics such as streaming ingestion, integration with SQL, handling latency, maintaining synchronization and deployment concepts [14]. After that, all these are analyzed and categorized into certain themes to explain present methods and changes in real-time data warehousing properly.

DATA ANALYSIS

Theme 1: Traditional batch-based ETL workflows lack responsiveness, leading to delays and reduced value in fast-paced business environments.

Most ETL run only at set times, so it takes a considerable amount of time for data to become ready for analysis [15]. Since action is needed, this time gap led to some situations being overlooked. Static pipelines may get congested when there are many tasks because they handle data all at once. Systems are hard to change, and adaptations are limited, while procedures are performed manually. It is important to have real-time data in finance, retail and logistics, as otherwise, businesses lose efficiency and face more risks. It becomes tough to upgrade old-style ETL pipelines with more and varied data arriving. While pipelines need to be updated, the data teams' workload goes up.

Companies find it challenging to make good decisions or plans because of missing real-time data. Such restrictions point out that the traditional batch approach is not well-suited for modern analytics [16]. Not delivering data in real time because of these ETL systems makes it harder for organizations to compete with others. Real-time processing needs to be introduced to help organizations eliminate their data silos, create more harmony and stay adaptable. Agile systems require immediate reaction to events so that they are aware in real time and can react faster. Programmers have to switch to better solutions because traditional approaches are not suitable for real-time data delivery.

Theme 2: In-memory and cloud-native platforms enhance real-time data ingestion, transformation, and analytics through high-speed processing and elasticity.

Data is processed in RAM, so that such platforms eliminate the delays created by disc-based storage. Such engines as Apache Spark and Apache Flink make it possible to transform and aggregate data from many sources more quickly. Snowflake, Databricks and Big Query are cloud-based platforms that boost this ability through automated scaling, dividing workloads and taking care of the infrastructure [17]. The systems make it possible to stream data continuously and query it rapidly with top-notch execution technology. Resource scalability ensures a steady output even when there is a high demand for services. Using automation tools and smooth connections, the process of creating pipelines becomes easier and less complicated. It can get real-time updates using dashboards, monitoring and alert systems without having to schedule them in batches. These technologies make sure that analytics reach users quickly and as expected by decision-makers. In-memory engines are fast, and cloud-native technologies ensure that applications can be used as needed [18]. Building systems for immediate data use is easy for data engineers. The operations in a business are supported by immediate feedback, easy-to-see information and reliable decisions. Data efficiency and scalability are in balance, this makes it easier to process data in real time.

Theme 3: Integrated data architectures bridge the gap between SQL querying and real-time streaming, supporting consistent and scalable analytics.

Using both SQL queries and fast streaming data to get consistent analytics across a large volume of data with integrated architectures. Counting on Lambda and Kappa architectures to balance the use of historical data and real-time feeds. Real-time engines handle new information as it arrives, but an SQL warehouse keeps saved data for analysis [19]. Data types are handled by both data engineers and analysts using the same standard SQL code with a single interface. Kafka, Flink and Delta Lake help to make data collection, processing and storage easy for all the platforms. Having a specific time for events, checkpoints, and managed schemas ensures there is no duplication or unusual changes. Integration frameworks make working with joins, aggregations and transformations in data, both static and streaming, much smoother. Making SQL compatible gives non-technical people better access and allows it to blend with existing BI applications. Using these architectures, it becomes possible for different data sources to merge and be managed within the same analysis area,

making things more flexible and secure [20]. Organizations are able to take in new data sets, sources and changes in rules without having to do huge restructuring. Being able to use both present and past data improves reporting, monitoring and creating projections. This system makes it possible for data warehousing to be flexible and connect to both business and analytics changes.

Theme 4: Synchronization, latency, and performance management remain critical in achieving seamless real-time data warehousing.

All parts, such as producers, processors and consumers, need to stay perfectly in sync to create a real-time data warehouse. The data must be delivered to the system precisely and properly because it is sent very fast. These tools limit data to a given time, check for duplicates and hold data until it is handled properly [21]. Improvement in latency through memory-based execution, vectorized calculations and smart query planning is possible. It tries partitioning, caching and using parallelization on clusters that are distributed to optimize performance. Keeping an eye on all data, hardware use and processing time to discover the problems that arise and ensure the way the load will be distributed. Different pipelines are brought together, and analyses are always performed consistently during continuous data ingestion with synchronization. These systems will be able to adjust to changes in schemas, transform the data and deal with delivery constraints instantly. High-quality system architecture greatly reduces the chances of any interruption or downtime while programmers query at the same time. Automation ensures that retries, recoveries after failures and limits on latency are handled automatically [22]. Current metrics are shown in real-time dashboards, that help operations monitor both the system and data. Scaling a system well and ensuring its reliability relies on carefully handling synchronization and performance controls. Enterprises gain actionable insights without sacrificing accuracy or availability. Data teams ensure real-time systems are efficient, help with making constant decisions and manage more complicated analytics.

Future Directions

The research will be helpful to incorporate AI-based automation into real-time data pipelines to reveal useful insights and find any anomalies. When using serverless data processing, teams need less time and effort to deploy applications and manage them. Looking more into decentralized designs and data mesh concepts encourages the ability of enterprise systems to scale [23]. Integration with real-time machine learning platforms allows adaptive decision-making. More people can access real-time analytics while low-code tools are reviewed. Proper investigation into compliance polices helps ensure handling of data with security and ethics. Business needs and advancements in technology are causing real-time data warehousing to change over time.

CONCLUSION

The study reviewed the way SQL data warehouses that focus on batch data processing have been replaced by integrated systems that analyse data in real time. The analysis brought out actions that achieved better performance, the working of synchronisation and changes in architecture to ensure continuous access to data. Having integrated frameworks allows users to easily seek information in both current and older data at high speeds and without limits. Leveraging in-memory systems and elastic architectures enables enterprises to enhance agility and support timely decision-making. Real-time data processing enhances cross-departmental efficiency and user engagement. Secondary qualitative analysis gave information on technology improvements. The findings of the research can help to make it possible to design data warehouses that can grow with the business, are flexible and deliver results in real time.

REFERENCES

[1]. Ali, A.R., (2018), March. Real-time big data warehousing and analysis framework. *In 2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA)* (pp. 43-49). IEEE.

[2]. Duggirala, S., (2018). Newsql databases and scalable in-memory analytics. *In Advances in Computers* (Vol. 109, pp. 49-76). Elsevier.

[3]. Gürcan, F. and Berigel, M., (2018), October. Real-time processing of big data streams: Lifecycle, tools, tasks, and challenges. *In 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)* (pp. 1-6). IEEE.

[4]. Malik, B.H., Maryam, M., Khalid, M., Khlaid, J., Rehman, N.U., Sajjad, S.I., Islam, T., Butt, U.A., Raza, A. and Nasr, M.S., (2019). *Fast and Efficient In-Memory Big Data Processing. International Journal of Advanced Computer Science and Applications*, 10(5).

[5]. Feinberg, B., (2019). Reducing data movement energy on dense and sparse linear algebra workloads: from machine learning to high performance scientific computing. *University of Rochester*.

[6]. Rajan, R.A.P., (2018), December. Serverless architecture-a revolution in cloud computing. *In 2018 Tenth International Conference on Advanced Computing (ICoAC)* (pp. 88-93). IEEE.
[7]. Fikri, N., Rida, M., Abghour, N., Moussaid, K. and El Omri, A., (2019). An adaptive and real-time based architecture for financial data integration. *Journal of Big Data*, 6, pp.1-25.
[8]. Jeon, Y.H., Lee, K.H. and Kim, H.J., (2019). Distributed join processing between streaming and stored big data under the micro-batch model. *IEEE Access*, 7, pp.34583-34598.
[9]. Thar, K., Oo, T.Z., Tun, Y.K., Kim, D.H., Kim, K.T. and Hong, C.S., (2019). A deep learning model generation framework for virtualized multi-access edge cache management. *IEEE Access*, 7, pp.62734-62749.
[10]. [10] Xing, B. and Marwala, T., (2018). The synergy of blockchain and artificial intelligence. *Available at SSRN 3225357*.
[11]. AL-Shboul, M.D.A., Garza-Reyes, J.A. and Kumar, V., (2018). Best supply chain management practices and high-performance firms: The case of Gulf manufacturing firms. *International Journal of Productivity and Performance Management*, 67(9), pp.1482-1509.
[12]. Braun, V. and Clarke, V., (2019). Reflecting on reflexive thematic analysis. *Qualitative research in sport, exercise and health*, 11(4), pp.589-597.
[13]. Eklund, M., (2018). Data Warehousing in the Cloud: Analysis of an Implementation Project.
[14]. Gianniti, E., (2018). Performance models, design and run time management of big data applications.
[15]. Machado, G.V., Cunha, Í., Pereira, A.C. and Oliveira, L.B., (2019). DOD-ETL: distributed on-demand ETL for near real-time business intelligence. *Journal of Internet Services and Applications*, 10(1), p.21.
[16]. Kolajo, T., Daramola, O. and Adebiyi, A., (2019). Big data stream analysis: a systematic literature review. *Journal of Big Data*, 6(1), p.47.
[17]. Zaidi, E., Thoo, E. and Heudecker, N., (2019). Magic quadrant for data integration tools. *Gartner Inc.*
[18]. Oliveira, M.P.V.D. and Handfield, R., (2019). Analytical foundations for development of real-time supply chain capabilities. *International Journal of Production Research*, 57(5), pp.1571-1589.
[19]. FG de Assis, L.F., EA Horita, F., P. de Freitas, E., Ueyama, J. and De Albuquerque, J.P., (2018). A service-oriented middleware for integrated management of crowdsourced and sensor data streams in disaster management. *Sensors*, 18(6), p.1689.
[20]. Ali, A.H., (2019). A survey on vertical and horizontal scaling platforms for big data analytics. International *Journal of Integrated Engineering*, 11(6), pp.138-150.
[21]. Pamisetty, A., (2019). Big Data Engineering for Real-Time Inventory Optimization in Wholesale Distribution Networks. *Available at SSRN 5267328.*
[22]. Kipf, A., Pandey, V., Böttcher, J., Braun, L., Neumann, T. and Kemper, A., (2019). Scalable analytics on fast data. *ACM Transactions on Database Systems (TODS)*, 44(1), pp.1-35.
[23]. Armbrust, M., Das, T., Torres, J., Yavuz, B., Zhu, S., Xin, R., Ghodsi, A., Stoica, I. and Zaharia, M., 2018, May. Structured streaming: A declarative api for real-time applications in apache spark. *In Proceedings of the 2018 International Conference on Management of Data* (pp. 601-613).
[24]. Armbrust, M., Das, T., Torres, J., Yavuz, B., Zhu, S., Xin, R., Ghodsi, A., Stoica, I. and Zaharia, M., 2018, May. Structured streaming: A declarative api for real-time applications in apache spark. *In Proceedings of the 2018 International Conference on Management of Data* (pp. 601-613).