Article

Not peer-reviewed version

Real-Time Big Data Technologies in Retail: Enhancing Personalization and Operational Efficiency

Addy Arif Bin Mahathir , Charan Teja Nagisettygari , Noor UL Amin * , Sai Rama Mahalingam , Sivamuganathan A/L Mohana Dass

Posted Date: 2 September 2025

doi: 10.20944/preprints202509.0257.v1

Keywords: big data technologies; real-time data processing; retail industry; apache hive; apache impala; apache spark; machine learning

 doi:10.20944/preprints202509.0257.v1

Article

Real-Time Big Data Technologies in Retail: Enhancing Personalization and Operational Efficiency

Addy Arif Bin Mahathir, Charan Teja Nagisettygari, Noor Ul Amin *, Sai Rama Mahalingam and Sivamuganathan A/L Mohana Dass

School of Computer Science, Taylor's University, Subang Jaya 47500, Malaysia

* Correspondence: nooraminnawab@gmail.com

Abstract

This study explores the integration of Big Data Technologies (BDT) in the retail industry, emphasizing their role in enabling real-time data processing and personalized customer experiences. The project examines how technologies such as Apache Hive, Impala, and Spark can process large-scale retail datasets to identify purchasing patterns, refine customer segmentation, and facilitate predictive analytics. The paper introduces four types of analytics—descriptive, predictive, prescriptive, and diagnostic—alongside machine learning algorithms, including supervised, unsupervised, and reinforcement learning, as well as Natural Language Processing (NLP). These tools collectively enable retailers to deliver dynamic, personalized marketing, enhance operational efficiency, and increase revenue. A practical experimentation using a Kaggle-based retail dataset evaluates the comparative performance of Hive, Impala, and Spark through SQL-like operations and MapReduce batch processing. Results show that while Impala excels in speed, Spark provides flexibility for complex data science tasks. The study concludes with an analysis of key considerations such as data storage, privacy, integration, and processing speeds necessary for effective big data deployment in retail environments.

Keywords: big data technologies; real-time data processing; retail industry; apache hive; apache impala; apache spark; machine learning

Introduction

"Big Data" is a term with many definitions, but at its core, it is data that is so large it surpasses that of a typical relational database in storage, processing, and analytics. Put differently, any question related to that data where the answer cannot be obtained using traditional database tools, then you have a Big Data challenge [1,2]. This report examines the use of Big Data technologies in the retail context. Specifically, it aims to chronicle how Big Data technologies are applied to improve retail operations - personalized marketing, pricing, customer engagement and inventory management. For example, to have personalized marketing campaigns based on customer interests. If a shopper frequently purchases toys, they can kick off a series of promotions directed only to toys and not toys and irrelevant products such as detergents [3–5].

With this research, we will explore how retailers use Big Data tools to enable retailers to offer a better shopping experience and improve their operational efficiency. By studying consumer behavior, paths to purchase, and product preferences, retailers can make better informed decisions, resulting in better margins, profits, and consumer experience. The subsequent sections of this report will go into detail about the specific technologies, analytics methods, and real-world examples that demonstrate how Big Data is revolutionizing retail[6–9].

 doi:10.20944/preprints202509.0257.v1

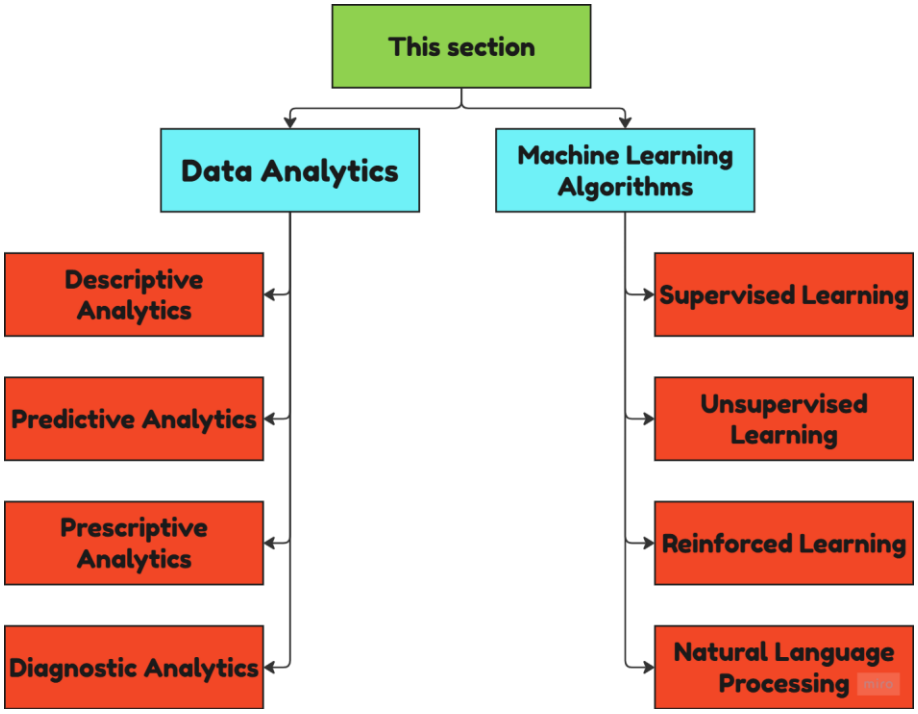

Figure 1. Summary of Data Analytics and Machine Learning Algorithms used in this research (Own work using Miro flowchart).

The figure above was created for the use of this paper only.

2. Literature Review

The use of Big Data Technologies in retail has positively disrupted conventional retail practices by allowing for real-time data handling, enhancing customer engagement and overall operational efficiencies. The literature illustrates a growing body of research that is uniting the distinct concepts of data analytics, machine learning, and distributed computing frameworks, in the growing use of Big Data in the retail context. Big Data is generally defined as large data sets that are too large to be handled using traditional database systems. The ability to process data in real-time is important in the retail world to remain competitive. Technologies like Apache Kafka and Hadoop provide companies the ability to capture, process and analyze massive data sets, giving organizations the framework to react in real-time to consumer behavior and any changes in the marketplace [10–14].

Real-time processing means that companies can implement personalized marketing practices, the key to driving customer interaction and engagement. Machine learning capability provides companies the ability to analyze historical purchase, browsing, and customer interaction to personalize recommendations and promotions [15–18]. AI-powered chatbots and virtual assistants further the enhanced customer experience by providing immediate and relevant responses directly tied to data-driven models[19–22].

Analytics is generally classified into four major types: (1) descriptive analytics that summarize what has occurred in the past; (2) predictive analytics that informs the behavior that is likely to occur in the future; and then prescriptive analytics that proposes actions based on a simulation that can be used for what-if scenarios; and (3) diagnostic analytics that provides cause and effect relationships [23–27]. These three analytics types allow retailers to make well-informed decisions, such as inventory and dynamic pricing decisions[28–30].

Machine learning algorithms (ML) are also differentiated by supervised or unsupervised algorithms— reinforcement learning, and natural language processing (NLP)—which are useful for customer segmentation, predicting churn, understanding sentiment, etc. Supervised models (e.g., decision trees, random forests) use labelled data to assist in classification or regression problems, while unsupervised models (e.g., the k-means clustering algorithm) have the ability to discover

 doi:10.20944/preprints202509.0257.v1

hidden patterns in unlabeled data[3]. Reinforcement learning focuses on maximizing long-term decisions, such as strategies that relate to stock replenishment; and NLP is used to enhance conversations and exploratory search through software applications such as chatbots [31,32].

Although considerations for software development exist, developing practical applications of analytics involves using computing technologies such as those offered by the Apache platform (Hadoop, Hive, Impala, Spark, etc.). Apache Hive is one of the products of the Apache Hadoop project and presents a SQL-like interface that provides access to data stored with the Hadoop data processing framework. It helps users to perform ETL operations, as well as build a data warehouse. Apache Impala - is a low-latency SQL execution engine for Hadoop that allows for interactive queries, and Apache Spark supports in-memory data processing capabilities with its libraries (such as MLlib and GraphX) for analytics operations - especially iterative operations [33–35]

An empirical comparative performance analysis using a real retail data set showed that Impala had the fastest execution time for batch processing (an average of 0.34s), whilst Hive was reasonable (3.21s) and Spark performed the slowest (12.25s). Implications included recognized trade-offs between low-latency queries (in Impala), large-scale data warehousing (in Hive), and advanced machine learning (in Spark)[36,37].

When dealing with Big Data, there are considerations regarding storage, processing speed, integration, and privacy. Storage came from the volume and variety of data that require cloud solutions and lifecycle management [37–39]. Despite the awareness of data privacy, which is ever-increasing in importance driven by regulatory requirements (e.g., GDPR; CCPA), retailers must take appropriate measures regarding secure and credible usage of data [40–42]. This is used with various applications [43–47], including the LTE and Cloud applications [48-49], as well as mainly earlier, and with current technological trends.

Methodology

This research provides a pragmatic and comparative assessment of the use and performance of BDTs in the retail industry, specifically in using real-time customer analytics and predicting customer spending patterns. The process includes technology selection, data preparation, implementation of data processing and machine learning algorithms, and performance evaluation based on a table of evaluation metrics.[48,49

1. Selection of Big Data Technologies

The three Big Data frameworks we chose to include in our study were selected due to their high-level use within industry and their respective processing functions:

- Apache Hive – for query against datasets within HDFS using SQL-like syntax with MapReduce.
- Apache Impala – for providing SQL query processing with low-latency, low-latency, high-speed processing of query statements across a distributed Hadoop ecosystem.
- Apache Spark – for distributed, in-memory processing of data stored across several distributed nodes and also for machine learning.

We ran our tools in both batch-processing and SQL-like environments to determine their efficiency and real-world usability with retail operations involving data.

2. Dataset Acquisition and Preprocessing

The dataset used in our study was publicly available through Kaggle, titled, “Customer Shopping Dataset – Retail Sales Data” by Aslan (2022). The dataset totaled over 99,000 customer records that were compiled from ten malls across Istanbul, including attributes like customer ID, gender, age, product category, payment method, price, quantity, and shopping mall name.

- Conversion: The dataset was initially in csv format, but we converted it to text format (.txt) to upload to Hadoop Distributed File System (HDFS).
- Categorization: Datasets were sorted into two categories:
- Demographics - customer status data (age, gender, customer id, etc.)
- Transaction - attribute data (price, quantity, category, method of payment, etc.).

 doi:10.20944/preprints202509.0257.v1

3. Experimental Design

Each Big Data technology was assessed through a series of structured tasks that echoed retail analytics applications. The tasks included:

- Batch Processing Tasks: We employed MapReduce frameworks within Hadoop to complete word count operations in order to determine credit card usage.
- SQL-Like Queries: We carried out five SQL-like tasks (e.g. top payment methods, top product categories, revenue computation) across Hive and Impala.
- In a non-SQL environment (Spark), a MapReduce pipeline was created using Java, and consisted of:
- Mapper Class – that tokenized the input data into key-value pairs.
- Reducer Class – that summarized mapped values.
- Driver Class – that controlled the execution on Spark Cluster.

4. Machine Learning Implementation

To support the deployment of predictive analytics, we implemented Machine Learning algorithms into Spark and a Python environment.

- Decision Tree Regression was implemented using Spark MLlib and Python (in a Jupyter Notebook) to predict customer spending behavior
- Feature Engineering: Data Preprocessing consisted of label encoding categorical variables, feature selection, and normalization.

Experimentation

This section discusses the Practical experimentation of three chosen Big Data Technologies that have been selected for this study namely Apache Hive, Impala and Spark.

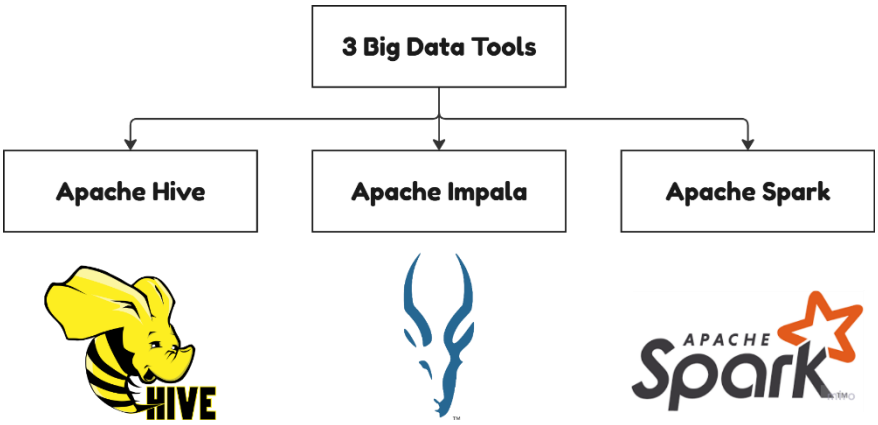

Figure 3. 3 Big Data tools for the practical experimentation in this section (Own work using Miro Flowchart).

The figure above was created for the use of this paper only.

It elaborates on their architecture diagram and how it is integrated into Hadoop HDFS, describes the metrics of for this experiment, finds a dataset to use for the experiment, tests it with these tools, puts the results into a table and then compares their performance to determine which technology performs best in which scenario. They also run a machine learning algorithm from the Big Data tools to predict customer preferences.

Evaluation Metrics

The performance of each technology was evaluated using three primary evaluation metrics. The first metric was execution time, which was assessed through the average time that each task took to complete for ten iterations to ensure consistency and accuracy in our results. The second metric was the accuracy of the results, which was evaluated through the reliability of the outputs that were established for each technology. For instance, one evaluation we performed was the amount of credit card users identified in our actions, which should be constant across all technologies, iterations, and

 doi:10.20944/preprints202509.0257.v1

runs. The last metric was ease of implementation, which was assessed qualitatively on syntax complexity, set-up and interactions with the technology while running and executing the tasks. Each of these evaluation metrics provided a basis for comparison and assessment of Hive, Impala, and Spark technologies used for data processing in a retail data environment.

Dataset

The dataset was maintained as comma separated values (.csv file). However, this file was converted to a text file to allow for an easier implementation during experimentation which was done using the three big data technologies described above. We will take time in the next section to discuss the matching between the new technologies.

Three Big Data Technologies

The "Customer Shopping Dataset - Retail Sales Data" dataset is the initial starting point for the actual retail technologies experimentation. Hence, the technologies have actual analytical tools in a real-time retail environment. The parameters are varied and complex enough to allow us to assess scalability and reliability of big data technologies that are designed for retail, including Hadoop (as well as Hive, Pig and Spark).

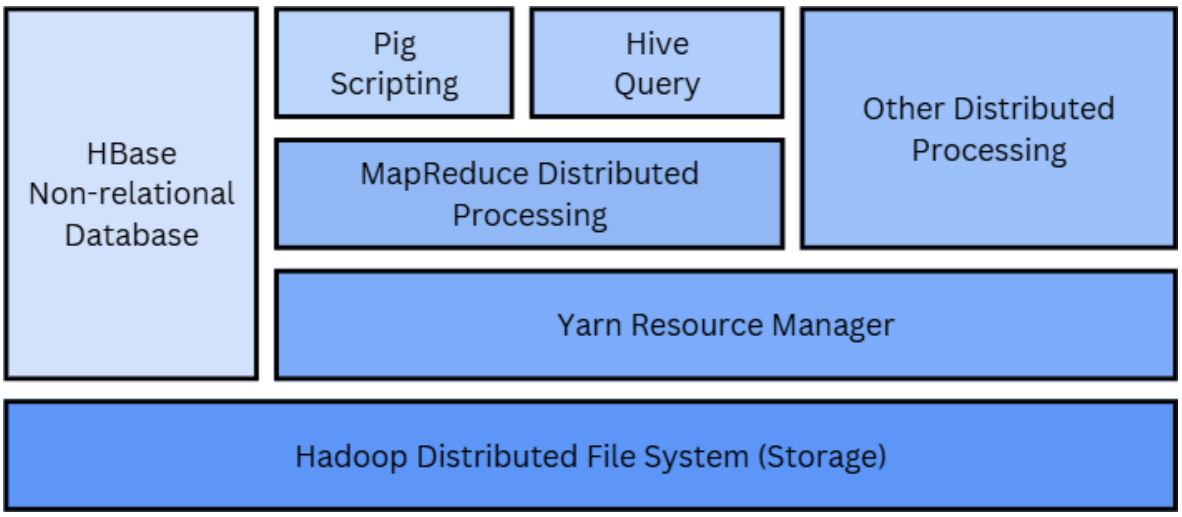

Figure 5. Apache Hadoop System Architecture.

Apache Hadoop serves as a free and open-source framework that is used for batch processing. The primary components allow large amounts of data to be stored and processed using Hadoop's Distributed File System (HDFS) for storage, and using Map Reduce for distributed processing. Apache Hadoop's architecture has 4 main layers: Distributed File System Layer, Resource Manager Layer, Processing Layer, and Data Management and Query Layer.

Hadoop uses a distributed architecture, also known as a master-slave architecture, for storage within the distributed file system layer. This architecture consists of two nodes: a master node and several slave nodes. The master node (Namenode) handles metadata processing, while slave nodes (DataNodes) store the data, as shown in Figure 6. The Hadoop architecture layers are followed by YARN (Another Resource Negotiator), which serves as a resource management layer that efficiently handles job scheduling. YARN also provides a flexible and scalable framework for running various distributed computing applications critical to the big data landscape. (Collins, 2023)

The next layer in the Hadoop architecture is the processing layer, called MapReduce. The processing layer consists of a method of running both iterative and structured operations while replicating other operations and can process large numbers of datasets in parallel. MapReduce consists of two functions: Map and Reduce. When the Map function is used, data is processed into key-value pairs. The Reduce function summarizes the results by aggregating them. Finally, the final layer in the Hadoop architecture is the data management and query layer, which is built using tools like Pig and Hive.

 doi:10.20944/preprints202509.0257.v1

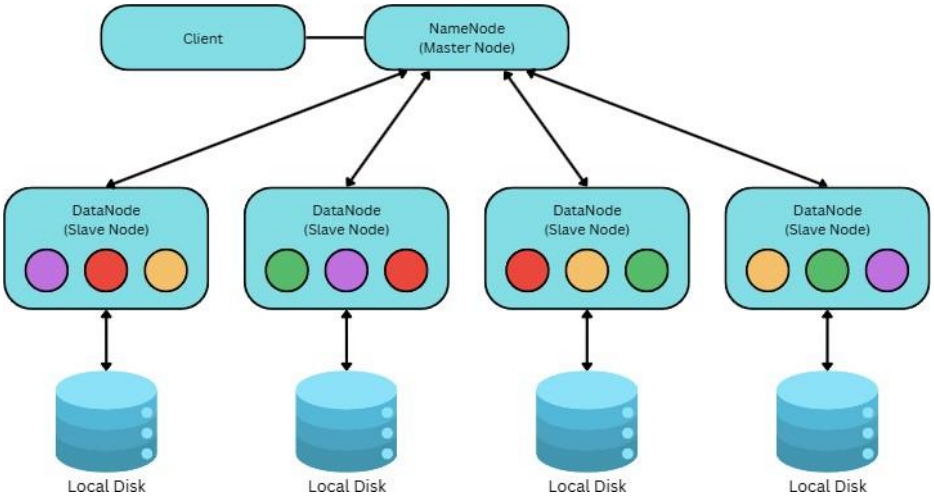

Figure 6. The Design of Hadoop Distributed File System (HDFS).

Hive

Apache Hive is a data warehouse and extract, transform, load (ETL) tool offering users a SQL-like interface to a Hadoop distributed file system (HDFS). Being built on top of Hadoop, it makes reading, writing and managing large datasets easier using Structure Query Language (SQL) structured syntax. Apache Hive is frequently used for common data warehousing tasks such as data encapsulation, ad hoc queries, and analysis of large datasets. Its primary focus is on scalability, extensibility, performance, fault-tolerance, and loose-coupling with input formats (GeeksForGeeks, 2018).

In Figure 7, there are five distinct facets in the overall structure of Apache Hive described as a system architecture: the User Interface (UI), Driver, Compiler, Metastore, and Execution Engine. The first item is the UI. The UI acts as an intermediary between the user and the system, allowing the user to submit queries or conduct other operations. The second aspect, which is known as the Driver, is the piece that accepts the queries and calls on the session handle, providing fetch and execute APIs through JDBC (Java Database Connectivity) and ODBC (Open Database Connectivity) interfaces.

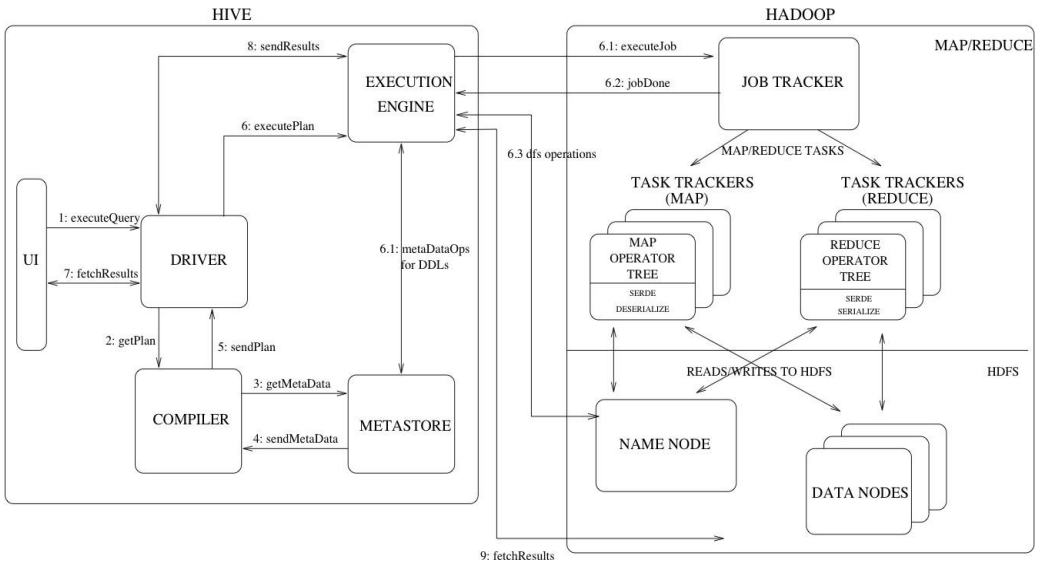

Figure 7. Apache Hive System Architecture (Apache Software Foundation, 2015).

The compiler is a component that parses the query and performs semantic analysis on the query blocks and expressions. The compiler then produces an execution plan based on the table and partition metadata received from the metastore. In addition, the metastore is the component that

 doi:10.20944/preprints202509.0257.v1

maintains all the structure information of the various tables and partitions within the warehouse. The metastore also includes such information as column, column type, serializers and deserializers to read and write, and the related HDFS files where the data is physically stored. Finally, the execution engine executes the execution plan produced by the compiler. It also takes care of the dependencies between the different steps, and executes them using the various system components.

Impala

Impala is an open source MPP (Massive Parallel Processing) SQL engine. Impala is designed to bring together two traditional pieces of an analytic database i.e. SQL support and translating data into meaningful insights with multi-user performance; to the scalability and flexibility of Apache Hadoop (Kornacker et al., n.d.). Plus, it also received the added benefits of production-grade security and administrative extensions from Cloudera Enterprise. There's no MapReduce either, Impala also has a specialized distributed query engine that is similar to the distributed query engine used in the commercial parallel Relational Database Management System (RDBMS); this is done to face what faced all other mass data systems: latency. The result is orders of magnitude faster performance than Hive depending on the type of query and its particular configuration (Apache Impala, n.d.).

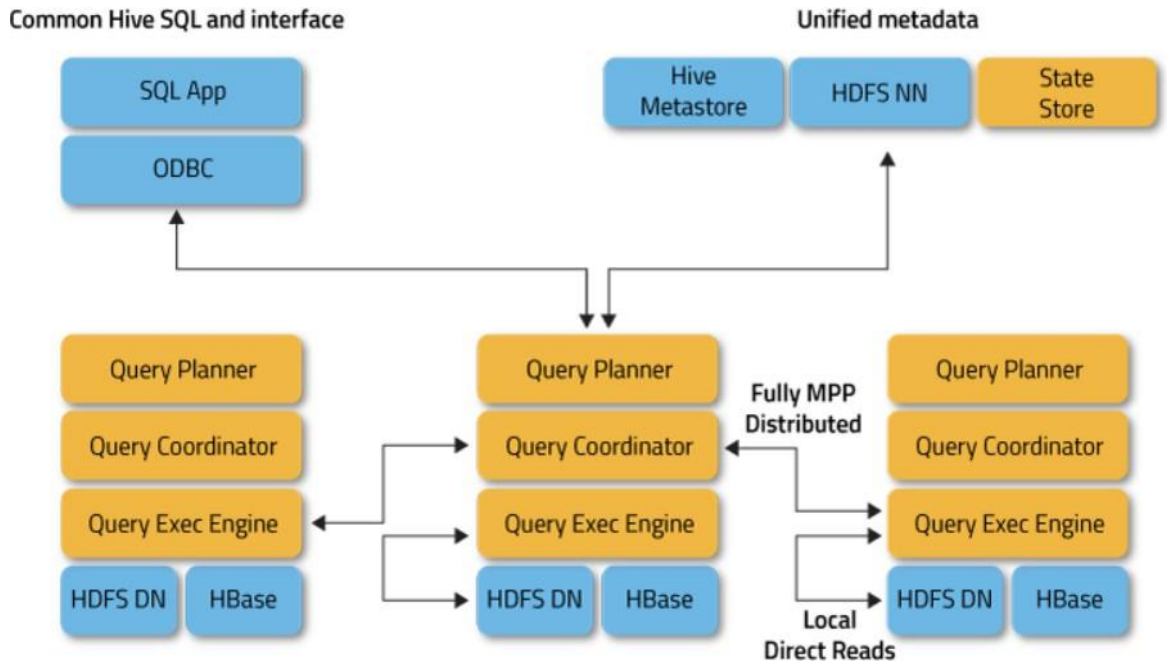

Figure 8. Apache Impala System Architecture (Apache Impala, n.d.).

An Apache Impala consists of three major components; Impala Daemon, Metastore, and Statestore. The Impala Daemon component runs on all nodes where Impala is installed. When a query is sent to an Impala Daemon on one of the nodes, that node becomes the "coordinator node" for that query. The Impala Daemon runs on all other nodes and can handle taking a number of queries at once. The Impala Daemon reads and writes to data files when it receives the query, and it also parallelizes queries by dividing the work amongst the Impala nodes within an Impala cluster. When the queries are being worked on by multiple Impala Daemon nodes, the results are sent back to the central coordinator.

The Statestore will check the health of each Impala Daemon node, and relay everything to all the other Daemons as fast as possible. In the event of a node failure for any reason, Statestore will relay that news to all other nodes. Upon receiving this information from the Statestore, other Daemons will refrain from assigning any more queries to the affected node. Impala utilizes a traditional query based databases for table definitions; thus, all the relevant information pertaining to table and column data and table definitions is stored in the Metastore. If a table definition or data is changed, all other Impala

 doi:10.20944/preprints202509.0257.v1

Daemons must refresh their metadata cache by getting the latest metadata, before issuing a new query against the intended table. (TutorialsPoint, 2024.

Spark

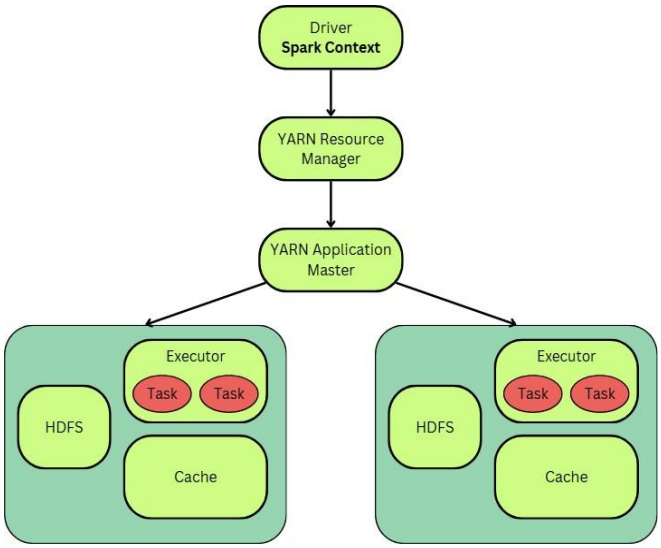

Figure 9. Apache Spark Distributed Processing Architecture.

The figure above was created for the use of this paper only.

Apache Spark was created to overcome some of the limitations of Hadoop technology. It uses a distributed system architecture, similar to Hadoop. Whenever we provide an input command to the system, the Driver program (which includes the Spark Context) sends Spark the details on how to access the cluster. Next, the Spark context sends instructions to the cluster manager to manage the jobs that run on the cluster. A job is made up of different tasks that are assigned to multiple slave nodes (CloudDuggu, 2023). In the architecture of Apache Spark Executors can access HDFS and Cache and use the cache to first retain the outputs in cache to speed up transferring data and reduce time lag.

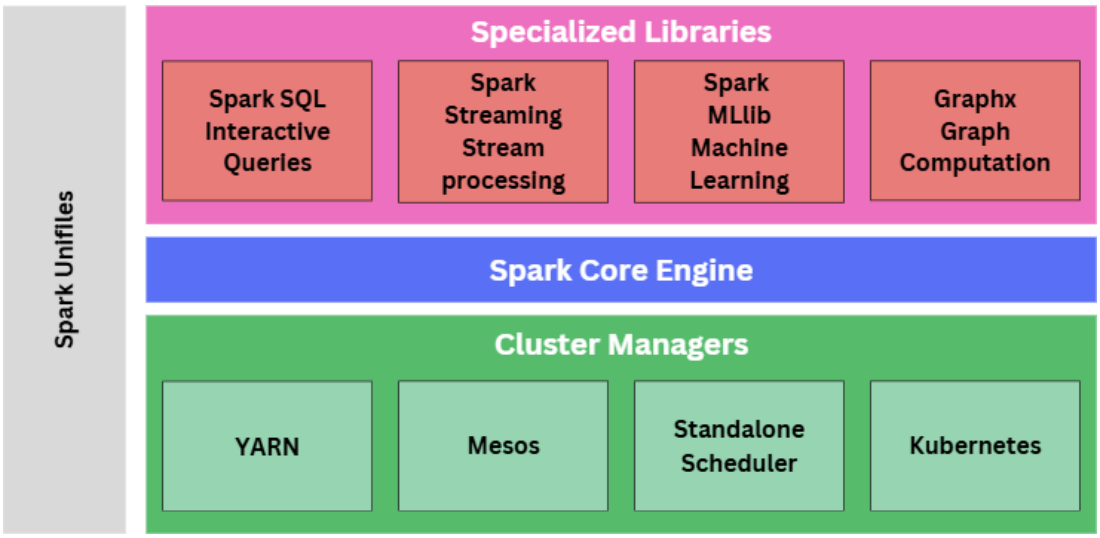

Figure 10. Apache Spark System Architecture.

The Spark framework is fundamentally built on the Spark Core Engine which supplies features such as job scheduling, memory management, fault recovery, and management of storage systems. This layer supports multiple storage systems like HDFS, Amazon S3, and others, making it ideal for

 doi:10.20944/preprints202509.0257.v1

various data sources. The function-specific libraries of Apache Spark provide ways to query structured data using SQL syntactic structure through Spark SQL, real-time data processing using Spark Streaming, distributed machine learning utilizing Spark MLlib, and graph-parallel and data-parallel processing utilizing GraphX. The Spark SQL library allows users to query and manage structured and semi-structured data using SQL syntactic structure and has a very efficient DataFrame API that abstracts the complexity of distributed processing so easier data analysis can be performed.

Because Spark exists "on top" of HDFS, it accesses Hive's SQL databases and allows the user to use SQL-like queries to analyse large data sets. These libraries are particularly good for data engineers and analysts who write SQL and want to take advantage of Spark's computing power for large data analysis.

Spark Streaming allows Spark to do real-time processing, making it useful for continuous data analysis applications. It allows Spark to stream and process realtime data by breaking the data streams into small micro-batches and simultaneously send to the Spark engine. With Spark Streaming, medical professionals can quickly turn streaming data into analysis, making it a perfect mix for the healthcare industry.

MLlib is used to quickly scale machine learning across large datasets in a distributed way. MLlib incorporates a variety of approaches and methods for classification, regression, clustering, collaborative filtering and dimensionality reduction. In addition, MLlib has methods for feature extraction, selection, and transformation, which enables data scientists to preprocess the data and develop machine learning pipelines. By using Spark, MLlib allows hospitals to implement machine learning at scale using distributed computing resources, and extract insights from large datasets from descriptive, predictive, prescriptive and diagnostics systems.

GraphX enables graph-based analysis and computation on a a large graph by providing an API for constructing, modifying, and processing graphs, and also allows for data-parallel and graph-parallel operations. It allows users to define graph analytics algorithms such as PageRank or connected components concisely, taking advantage of Spark's distributed processing capabilities. This library is extremely useful for assessing relationships in data and structures, making it well suited for healthcare applications such as social network analysis and any other scenarios where complex relationships and networks make sense.

Lastly, Resilient Distributed Datasets (RDDs) are distributed data structures that allow parallel operation and enable fault tolerance. RDDs introduced by Spark allowed for flexible and reliable large scale data processing with the ability to easily manipulate data across clusters and allow for transformations and processing to take place within an implementation of Apache Spark.

Comparative Analysis

In the end, we put all results in table and visuals in order to compare and contrast strengths and weaknesses of Hive, Impala and Spark in terms of speed, scalability, and usability for retail analytics. I reported and plotted the shortest, longest, and average processing times so that we could form conclusions about the choices an enterprise may take when making a technology choice.

Discussion

The present project investigated the integration and effectiveness of Big Data Technologies (BDT) within the retail sector relating to real-time data processing, customer experience improvements, and comparison between three of the leading tools: Apache Hive, Impala and Spark. The findings provide a valuable overview of how retail companies can leverage big data better in supporting decision-making, operational efficiencies, and customer service personalization.

The analysis began by outlining the current conceptualizations of how Big Data and the use of real-time processing tools like Apache Kafka and Hadoop impact and change retail practices, dynamic inventory management, personalized marketing, and interactive customer service using AI chat-bots. The project evidenced that real-time data, paired with machine learning algorithms, is significant in spotting trends and customer behaviors for optimum advertising and delivery of services.

 doi:10.20944/preprints202509.0257.v1

The discussion then delved into various analytics types—descriptive, predictive, prescriptive, and diagnostic—and machine learning models, including supervised, unsupervised, reinforcement learning, and NLP. These tools collectively allow retailers to track customer habits, forecast future demands, suggest optimal decisions, and understand consumer sentiments.

We utilized a real-world dataset from Kaggle to evaluate operational performance of Apache Hive, Impala and Spark. The assessment criteria were execution time, implementation complexity, and result accuracy based on the processing time of Impala, Hive, and Spark during tasks that measured performance. Impala outperformed the other systems--average batch processing time= 0.34 seconds when comparing to Apache Hive (average=3.21 seconds) and Spark (average= 12.25 seconds). This means Impala is good selection for querying data needing low latency and providing answers quickly.

Hadoop Hive was not as fast as Impala but it does work well for complex SQL-like queries and easy interface with Hadoop's HDFS and is better option for structured query over a substantially larger space. Spark performed poorly for batch tasks but delivered all functionality to support distributed machine learning tasks in regard to real-time stream processing for advanced analytics. Apache Spark is better choice when working with streaming processing and execution of large scale model (machine learning) pipelines. Also, we see from SQL-type operations that both Hive and Impala can support conventional database-style requests within a big data context respectively. The short representative tasks of identifying most used payment methods, highest selling product categories, and revenue per mall have indicated that SQL structured query languages can be impactful and relevant under big data context.

Overall, the research presented some other to areas reconsiderations to ensure successful deployment of big data in the retail context, such as the need for scalable storage for BI environments, robust privacy considerations from the onset (under GDPR, CCPA), fast processing time, and in particular managing all relevant data from sources of large variety.

As a conclusion, this research has also provided validation that the use of appropriate big data technologies should provide a decisive competitive advantage within retail procurement decision making. Where there are appropriate use cases for each technology, overall benefits are maximized when the technology is applied to the relevant operational context, with Impala representing speed, Spark representing machine learning/real time applications, and Hive representing big data batch processing using SQL-like commands.

Conclusion

This study documented our investigation into the impact of Big Data Technologies on transforming the retail marketplace by utilizing real-time processing, personalized information for customers, and predictability of insights from advanced analytics and machine learning algorithms. By measuring the deployment and comparison of Hive, Impala, and Spark, we had the opportunity to clarify the ability and performance characteristics of each technology regarding large-scale tasks relevant to radical efficiency in retail. Impala was the fastest technology for batch processing because of its ability to provide low-latency performance for ad hoc queries as well as batch-processing data. Spark was a great option for processing complex data because of its in-memory computing, but it had been proven slower than Impala in the current experiment. Hive was good and reliable option but not as fast as Impala.

This study documented our investigation into the impact of Big Data Technologies on transforming the retail marketplace by utilizing real-time processing, personalized information for customers, and the predictability of insights from advanced analytics and machine learning algorithms. By measuring the deployment and comparison of Hive, Impala, and Spark, we had the opportunity to clarify the ability and performance characteristics of each technology regarding large scale tasks relevant to radical efficiency in retail. Impala was the fastest technology for batch processing because of its ability to provide low-latency performance for ad hoc queries as well as batch-processing data. Spark was a great option for processing complex data because of its in-

 doi:10.20944/preprints202509.0257.v1

memory computing, but it had been proven slower than Impala in the current experiment. Hive was a good and reliable option, but not as fast as Impala.

References

1. Alkan, N., Menguc, K. and Kabak, Ö. (2022). Prescriptive Analytics: Optimization and Modeling. Springer Series in Advanced Manufacturing, [online] pp.239–264. doi:https://doi.org/10.1007/978-3-030-93823-9_9.
2. Ankur, S. (2023). The Top Distributed Data Processing Technologies: A Comprehensive Overview. [online] Medium. Available at: https://medium.com/@singhal.ankur8/the-top-distributed-data-processing-technologies-a-comprehensive-overview-712756db3242.
3. Apache Impala (n.d.). Impala. [online] impala.apache.org. Available at: https://impala.apache.org/overview.html.
4. Apache Software Foundation (2015). Design - Apache Hive - Apache Software Foundation. [online] cwiki.apache.org. Available at: https://cwiki.apache.org/confluence/display/hive/design.
5. Apache Software Foundation (2019). Apache Hadoop. [online] Apache.org. Available at: https://hadoop.apache.org/.
6. Aslan, M.T. (2022). Customer Shopping Dataset - Retail Sales Data. [online] www.kaggle.com. Available at: https://www.kaggle.com/datasets/mehmettahiraslan/customer-shopping-datas et [Accessed 6 Nov. 2024].
7. Azhar, A. (2022). What is data integration? | TechRepublic. [online] TechRepublic. Available at: https://www.techrepublic.com/article/data-integration/ [Accessed 29 Oct. 2024].
8. Bajaj, P., Ray, R., Shedge, S., Vidhate, S. and Shardoor, N. (2020). SALES PREDICTION USING MACHINE LEARNING ALGORITHMS. [online]
9. International Research Journal of Engineering and Technology. Available at: https://www.academia.edu/download/64640730/IRJET-V7I6676.pdf [Accessed 29 Oct. 2024].
10. Capriolo, E., Wampler, D. and Rutherglen, J. (2012). Programming Hive. O'Reilly Online Learning. O'Reilly Media, Inc.
11. Collins, N. (2023). What Is YARN In Big Data | Robots.net. [online] Robots.net. Available at: https://robots.net/fintech/what-is-yarn-in-big-data/.
12. Cote, C. (2021). What Is Descriptive Analytics? 5 Examples | HBS Online. [online] Business Insights - Blog. Available at: https://online.hbs.edu/blog/post/descriptive-analytics [Accessed 31 Oct. 2024].
13. Dutta, S. (2024). What is Data Integration and Importance of Data Integration. [online] www.sprinkledata.com. Available at: https://www.sprinkledata.com/blogs/importance-of-data-integration.
14. Favaretto, M., De Clercq, E., Schneble, C.O. and Elger, B.S. (2020). What is your definition of Big Data? Researchers' understanding of the phenomenon of the decade. PLOS ONE, 15(2), p.e0228987. doi:https://doi.org/10.1371/journal.pone.0228987.
15. GeeksForGeeks (2018). Apache Hive. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/apache-hive/.
16. GeeksForGeeks (2019). Introduction to Apache Pig. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/introduction-to-apache-pig/ [Accessed 18 Dec. 2024]. Last Updated : 14 May, 2023.
17. GeeksforGeeks (2023). Machine Learning Algorithms. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/machine-learning-algorithms/ [Accessed 5 Nov. 2024].
18. Grolinger, K. and AlMahamid, F. (2022). Reinforcement Learning Algorithms: An Overview and Classification. [online] Arxiv. Available at: https://arxiv.org/pdf/2209.14940 [Accessed 15 Dec. 2024].
19. ishrakhussain (2023). EDA and Prediction for customer spending. [online] Kaggle.com. Available at: https://www.kaggle.com/code/ishrakhussain/eda-and-prediction-for-customer- spending/notebook [Accessed 21 Dec. 2024].
20. Kalyanathaya, K.P., Akila, D. and Rajesh, P. (2019). Advances in natural language processing–a survey of current research trends, development tools and industry applications. International Journal of Recent Technology and Engineering, 7, pp.199–202.

21. Kelly, B. (2023). Positive Retail | Cloud vs On-premise POS. [online] Positive Retail. Available at: https://positiveretail.ie/cloud-vs-on-premise-pos/ [Accessed 29 Oct. 2024].
22. Khurana, D., Koli, A., Khatter, K. and Singh, S. (2022). Natural Language processing: State of the art, Current Trends and Challenges. Multimedia Tools and Applications, 82(3), pp.3713–3744. doi:https://doi.org/10.1007/s11042-022-13428-4.
23. Kornacker, M., Behm, A., Bittorf, V., Bobrovytsky, T., Ching, C., Choi, A., Erickson, J., Grund, M., Hecht, D., Jacobs, M., Joshi, I., Kuff, L., Kumar, D., Leblang, A., Li, N., Pandis, I., Robinson, H., Rorke, D., Rus, S. and Russell, J. (n.d.). Impala: A Modern, Open-Source SQL Engine for Hadoop. [online] Available at: https://www.cidrdb.org/cidr2015/Papers/CIDR15_Paper28.pdf.
24. Krissberg, C. (2024). Cloud vs On Premise Infrastructure: What's Best for Your Business IT Operations. [online] www.linkedin.com. Available at: https://www.linkedin.com/pulse/cloud-vs-premise-infrastructure-whats-best-yo ur-cody-krissberg-ona9e/.
25. Lawton, G. (2022). What is Descriptive Analytics? Definition from WhatIs.com. [online] WhatIs.com. Available at: https://www.techtarget.com/whatis/definition/descriptive-analytics [Accessed 31 Oct. 2024].
26. Maceira, J. (2024). Cloud Platforms vs. On-Premise Solutions Which one to choose? [online] Orienteed • leading e-commerce solutions. Available at: https://orienteed.com/en/cloud-platforms-vs-on-premise-solutions/ [Accessed 29 Oct. 2024].
27. Moesmann , M. and Pedersen, T.B. (n.d.). ata-Driven Prescriptive Analytics Applications: A Comprehensive Survey. Arxiv. [online] doi:https://arxiv.org/html/2412.00034v1.
28. Mohan, V. (2022). CCPA vs GDPR: The 5 Differences You Should Know. [online] Sprinto. Available at: https://sprinto.com/blog/ccpa-vs-gdpr/.
29. Ouaknine, K., Carey, M. and Kirkpatrick, S. (2015). The PigMix Benchmark on Pig, MapReduce, and HPCC Systems. doi:https://doi.org/10.1109/bigdatacongress.2015.99.
30. OpenAI. (2024). ChatGPT (Dec 14 version) [Large language model]. Available at: https://chatgpt.com/share/675d985b-5b10-8005-8aa2-4a3226a599a6.
31. Saeed, S. (2019). Analysis of software development methodologies. IJCDS. Scopus; Publish.
32. Saeed, S. (2019). The serverless architecture: Current trends and open issues moving legacy applications. IJCDS. Scopus.
33. Saeed, S., & Humayun, M. (2019). Disparaging the barriers of journal citation reports (JCR). IJCSNS: International Journal of Computer Science and Network Security, 19(5), 156-175. ISI-Index: 1.5.
34. Saeed, S. (2016). Surveillance system concept due to the uses of face recognition application. Journal of Information Communication Technologies and Robotic Applications, 7(1), 17-22.
35. Shi, J.-C., Yu, Y., Da, Q., Chen, S.-Y. and Zeng, A.-X. (2019). Virtual-Taobao: Virtualizing Real-World Online Retail Environment for Reinforcement Learning. Proceedings of the AAAI Conference on Artificial Intelligence, [online] 33(01), pp.4902–4909. doi:https://doi.org/10.1609/aaai.v33i01.33014902.
36. Stedman, C. (2023). What is Data Analytics? - Definition from WhatIs.com. [online] SearchDataManagement. Available at: https://www.techtarget.com/searchdatamanagement/definition/data-analytics [Accessed 4 Nov. 2024].
37. Tucci, L. (2021a). What Is Machine Learning and Why Is It Important? [online] Techtarget. Available at: https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML [Accessed 5 Nov. 2024].
38. Tucci, L. (2021b). What is Predictive Analytics? An Enterprise Guide. [online] SearchBusinessAnalytics. Available at: https://www.techtarget.com/searchbusinessanalytics/definition/predictive-anal ytics [Accessed 4 Nov. 2024].
39. Turkmen, B. (2022). Customer Segmentation with Machine Learning for Online Retail Industry. The European Journal of Social & Behavioural Sciences, [online] 31(2), pp.111–136. doi:https://doi.org/10.15405/ejsbs.316.
40. TutorialsPoint (2024). Impala - Architecture. [online] Tutorialspoint.com. Available at: https://www.tutorialspoint.com/impala/impala_architecture.htm [Accessed 20 Dec. 2024].

 doi:10.20944/preprints202509.0257.v1

41. Yasar, K. (2023). What is Data Analytics? - Definition from WhatIs.com. [online] SearchDataManagement. Available at: https://www.techtarget.com/searchdatamanagement/definition/data-analytics [Accessed 31 Oct. 2024].
42. Miro. (n.d.). Flowchart maker. Miro. https://miro.com/flowchart/
43. Jhanjhi, N.Z. (2025). Investigating the Influence of Loss Functions on the Performance and Interpretability of Machine Learning Models. In: Pal, S., Rocha, Á. (eds) Proceedings of 4th International Conference on Mathematical Modeling and Computational Science. ICMMCS 2025. Lecture Notes in Networks and Systems, vol 1399. Springer, Cham. https://doi.org/10.1007/978-3-031-91005-0_43
44. Humayun, M., Khalil, M. I., Almuayqil, S. N., & Jhanjhi, N. Z. (2023). Framework for detecting breast cancer risk presence using deep learning. Electronics, 12(2), 403.
45. Gill, S. H., Razzaq, M. A., Ahmad, M., Almansour, F. M., Haq, I. U., Jhanjhi, N. Z., ... & Masud, M. (2022). Security and privacy aspects of cloud computing: a smart campus case study. Intelligent Automation & Soft Computing, 31(1), 117-128.
46. Aldughayfiq, B., Ashfaq, F., Jhanjhi, N. Z., & Humayun, M. (2023, April). Yolo-based deep learning model for pressure ulcer detection and classification. In Healthcare (Vol. 11, No. 9, p. 1222). MDPI.
47. N. Jhanjhi, "Comparative Analysis of Frequent Pattern Mining Algorithms on Healthcare Data," 2024 IEEE 9th International Conference on Engineering Technologies and Applied Sciences (ICETAS), Bahrain, Bahrain, 2024, pp. 1-10, doi: 10.1109/ICETAS62372.2024.11119839.
48. Ray, S. K., Sirisena, H., & Deka, D. (2013, October). LTE-Advanced handover: An orientation matching-based fast and reliable approach. In 38th annual IEEE conference on local computer networks (pp. 280-283). IEEE.
49. Samaras, V., Daskapan, S., Ahmad, R., & Ray, S. K. (2014, November). An enterprise security architecture for accessing SaaS cloud services with BYOD. In 2014 Australasian Telecommunication Networks and Applications Conference (ATNAC) (pp. 129-134). IEEE.