Enhancing Decision-Making in IoT Ecosystems with Big Data Analytics and Hadoop Frameworks

Anu Vij[1*]
Department of Computer Science & IT, GNDU University, Amritsar, Punjab, India, *vijanu370@gmail.com*

Aniket Goyal[2]
Department of Computer Science Electrical, MP Research Work Mathura, Uttar Pradesh, India, *aniketgoyal@mpsquare.in*

Abstract:

Background: Decision-making in IoT ecosystems involves using data from interconnected devices to make real-time, informed decisions that improve efficiency and functionality. This research tackles the significant challenge of real-time decision-making in Internet of Things (IoT) ecosystems by integrating Big Data Analytics (BDA) and Hadoop frameworks. This study aims to develop and assess a sophisticated decision-making model that utilizes BDA and Hadoop to boost operational efficiency, predictive maintenance, and actionable insights in IoT settings.
Methods: The methodology includes employing Python libraries for data analysis, Hadoop Distributed File System (HDFS) for scalable storage, and Hive for structured querying. Key evaluation parameters are latency, throughput, delay metrics, regression predictions, and K-means clustering for anomaly detection.
Results: The findings reveal substantial enhancements, with the average delay reduced to approximately 7.1 ns, a maximum delay of 9.31 ns, and a minimum delay of 2.08 ns. Throughput values varied between 0.5 and 2 samples/second, showcasing efficient processing capabilities. Regression analysis and K-means clustering successfully identified significant delays and anomalies, proving the model's efficacy.
Conclusion: This research offers scalable and fault-tolerant solutions for real-time data processing in IoT ecosystems. Future research should focus on refining predictive analytics, optimizing data processing frameworks, and exploring hybrid systems to ensure secure data management.

Keywords: Big Data Analytics (BDA), Decision-Making, Hadoop Framework, IoT Ecosystems, Machine Learning

1. Introduction

The Internet of Things (IoT) is a new concept that changes the optics of activities in many areas, including healthcare production, the development of smart cities, and engagement in agriculture (Nathali Silva et al., 2017). The IoT environment is represented by intelligent devices and sensors that constantly produce massive quantities of data. This has enabled the users to have the concept of real-time conditions to allow for automated processes of decision-making (Bibri, 2018; Nisar et al., 2021; Hu & Shu, 2023). However, the data challenge: the quantity, variety, and velocity of data generated from the IoT devices is one of the data management and decision-making frameworks many traditional frameworks grapple with (Hussain et al., 2023). Processing and analyzing the data becomes harder in real time as the systems get more complicated and extensive chart systems (Osman, 2019). Most often, this results in allocation and waste of time, which is a barrier to the organized efforts for IoT to be able to generate real-time intelligence. In this respect, efficient streaming and analysis of data presents the greatest challenge in terms of innovation and searching for solutions, hence Big Data Analytics (BDA) (Li, 2020).

BDA is the central enabler of the success of IoT systems because it elicits significant insights from the vast amounts of data generated (Shahat Osman & Elragal, 2021). Technologically, issues in storing, processing, and deriving actionable intelligence from heterogeneous sources of data haunt IoT systems (Silva et al., 2018). BDA, therefore, offers tools and technologies to address these challenges by providing scalable ways to manage complex datasets (Arora et al., 2023; Al-Jumaili et al., 2023). With these advanced methods, BDA can assist in IoT data analysis by discovering patterns and correlations that could be used as decisions. In this case, smart cities are a good example: BDA could facilitate traffic management, energy consumption, and public safety optimization, using data processing of sensors, cameras, and other IoT devices in real-time (Mukherjee et al., 2022; Vashishth et al; Demertzis et al., 2020). BDA transforms raw data into actionable insights that enhance the IoT ecosystem's capabilities to make smart, timely, and automated decisions (Allam, 2017; Zineb et al., 2021).

Along with BDA, Hadoop is another very handy tool regarding data management and analytics. Hadoop is an open-source framework that supports a very widely distributed environment for the storage and processing of large amounts of datasets across multiple computers (Koren et al., 2019; Liu et al., 2021). This framework is quite suitable for the IoT ecosystems as it can be scalable, tolerant of faults, and able to handle heterogeneous data easily. IoT devices generate structured, semi-structured, and unstructured data that can be stored and processed efficiently using Hadoop Distributed File System (HDFS) and the associated MapReduce processing model (Paramesha et al., 2024; Rehman et al., 2022). That is, Hadoop's distributed nature makes sure data processing occurs in parallel on multiple nodes; thus, the time to analyze large datasets is quite less. Additionally, being fault-tolerant, the availability of data and continuous processing ensure data remains accessible even after the failure of some nodes. Thus, it is a reliable solution to the growing demands of IoT data management. Figure 1 depicts the basic collaboration between the Hadoop Framework and Big Data technologies (Rossi & Bianchi, 2022; Ibrahim & Aoun, 2022).

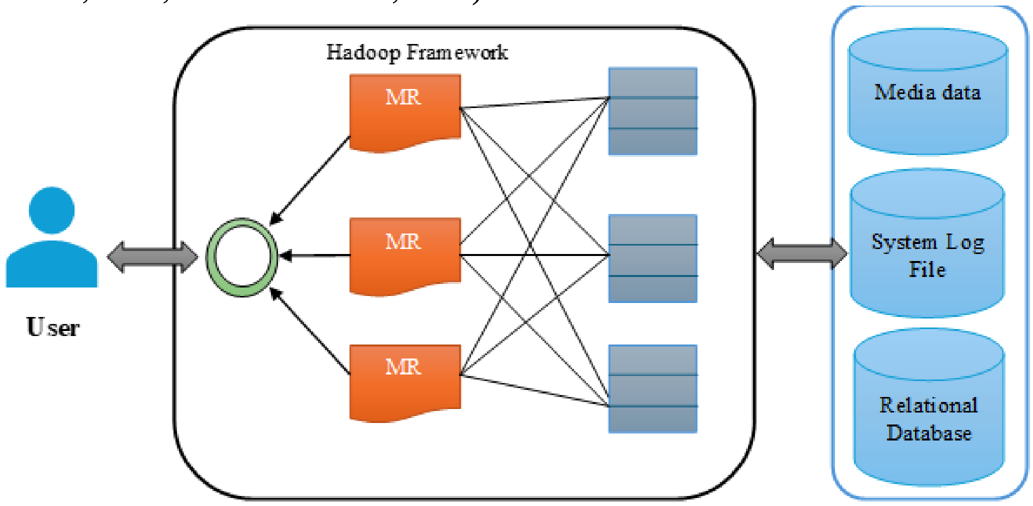

Figure 1: Basic collaboration among MapReduce, Hadoop, and Big Data (Desarkar & Das, 2017)

The integration of BDA and Hadoop frameworks allows companies to leverage the full value of IoT ecosystems by overcoming some of the key challenges related to data management, processing, and analysis (Nguyen et al., 2022). Such technologies form the backbone of decision-making across industries, especially by efficiently managing large datasets, extracting actionable insights, and basing decisions in real-time (Ikegwu et al., 2022; Sharma & Barua, 2023). With this evolution and growth of the IoT ecosystem, the BDA-Hadoop combined approach is becoming increasingly important to help organizations realize the value they expect to gain from their investments in IoT through data-driven decisions to drive innovation. Altogether, they offer an end-to-end solution that can benefit operational efficiency along with the creation of a smart and responsive IoT system fitting an interconnected world (Palanisamy & Thirunavukarasu, 2019).

This research aims at designing and evaluating an integrated decision-making model through Big Data Analytics and Hadoop frameworks to leverage optimization in the decision process within IoT ecosystems, thereby providing real-time data-driven insights for improved operational efficiency, predictive maintenance, and better decisions across various IoT applications.

The scope of this research involves the design and evaluation of a Big Data Analytics-driven decision-making model within IoT ecosystems. It covers the integration of IoT data sources, the development of Hadoop-based frameworks for data management, and the optimization of real-time decision-making processes, aiming to enhance efficiency and actionable insights across IoT applications.

The significance of this research lies in addressing the challenges of IoT data management by introducing an advanced decision-making model powered by Big Data Analytics and Hadoop frameworks. By efficiently managing vast IoT data, the model enhances operational efficiency, predictive capabilities, and actionable insights, contributing to smarter, more responsive IoT applications across various industries.

This research has the following contributions:

- This research contributes to the development of a Big Data Analytics and Hadoop-based decision-making model that enhances real-time decision processes in IoT ecosystems.
- The study applies Hadoop frameworks to efficiently manage and process large-scale IoT data, improving operational efficiency and predictive capabilities.
- The research addresses the challenges of managing massive, diverse IoT data by providing a scalable, fault-tolerant solution that ensures actionable insights for improved decision-making.
- The model is rigorously evaluated for its effectiveness in optimizing IoT ecosystem performance, ensuring reliable, data-driven insights across various IoT applications.

Section 1 of this research presents an overview of the subject matter. Section 2 delineates the pertinent contributions of numerous researchers. Section 3 delineates the proposed methodology. Section 4 presents the results and discussion. Section 5 presents the conclusion and future directions for research.

2. Review of Literature

This section carries an integrative review of the studies of relevant authors based upon Enhancing Decision-Making in IoT Ecosystems with Big Data Analytics and Hadoop Frameworks.

Haddad et al., (2024) developed a cutting-edge methodology for sentiment prediction with deep learning, batch processing, and streaming BDA. Researchers utilized distributed systems such as Hadoop and Spark for stream preparation and involved preliminary data cleansing, volume reduction, minimizing time to access, and minimizing volume storage. The research's big data datasets were processed from brief volume scripts using batch and streaming frameworks with deep learning implemented for Natural Language Processing (NLP). The experimental results validated the efficacy of the proposed model, reaching 96% accuracy, and also demonstrated its superiority to existing methods in the literature.

Hasanpuri et al., (2024) suggested an advanced distributed cluster-based system for performance evaluation of extensive IoT datasets utilizing Big Data analytics tools to tackle issues in the processing and analysis of IoT data. It proposed a scalable and fault-tolerant methodology for evaluating the IoT analytics system. Important contributions and applications were highlighted, thus advancing the research on IoT data analytics. The framework proved to be a useful tool for businesses and scientists working with large IoT data sets and enabled many insights to be gained. The Test of MapReduce for TeraSort operation showed unique throughput performance, which decreased as data size went beyond 200GB, stressing the limitations in performance due to the increase in IoT data.

Thanekar and Puri (2024) proposed a metadata-driven safe approach using Hadoop to enhance data processing as well as storage with reduced unnecessary data transit and job execution time. This research enhanced the understanding and the real-world application of deduplication in Hadoop systems as well as provided valuable insights to scholars and practitioners dealing with large data processing.

Patidar et al. (2024) carried out Hadoop-oriented Large-Scale Attacks Data Analytics, seeking to bypass some elementary security vulnerabilities in the maturing IoT space. The study looked at some of the threats that the IoT devices were subjected to using the large-scale and novel dataset CICIoT2023. Some of the researchers processed the data using the Big Data technologies Hadoop and Hive, as well as the visualization tools Microsoft Power BI to derive actionable insights from complex data. Such insights emphasize the weaknesses and the attacks within the IoT ecosystem.

Rahmani et al. (2024) implemented a Workflow Scheduler Based On the Hadoop Framework (WSH), aware of heterogeneity in the system while planning jobs that were to be executed with high processing power or Input/Output (IO). Before task scheduling, WSH gathered information through a training task. The results show better utilization of resources and a smaller makespan due to load balancing in job allocation in Hadoop. It was demonstrated, using a variety of workflows and experimental data, that the proposed method outperformed the algorithm in terms of the scheduling length ratio (42% improvement), makespan (20% reduction), and speedup (around 37%).

Fatima et al. (2023) investigated the application of the recent development of the big data paradigm in distributed and parallel computation focusing on the agriculture industry. In an attempt to enhance farming methods and consideration concerning facts, it was sought to investigate the frameworks such as Hadoop and Spark used in the context of agricultural data collection, storage, processing, and analysis. It became apparent that farming operations and their decision-making processes improved significantly due to the capabilities of Hadoop and Spark frameworks towards agricultural data processing and analysis. This research has elaborated on the potential of big data analytics in the agricultural sector, including the aspects of distributed computing and parallel processing.

Demirbaga et al. (2022) developed a big data architecture level health data analysis solution with an IoT application, which helps to demonstrate how verifiable access to resources should be done. The architecture consists of 2 subsystems: the BDA monitoring system and the blockchain data storage and access system. This method combined big data and blockchain for the secure storage and analytics of IoT data, thus validating its access using a zero-knowledge protocol to secure illegal access to the information and data likability. The overall findings suggest that a method resulted in efficiency regarding solving BDA and privacy problems in healthcare. For 100 users, mining time was reported as 289 seconds, and the same for 200 users was 358 seconds.

Pajooh et al., (2021) designed a layer-based distributed data storage architecture for a blockchain-enabled large-scale IoT system and implemented it using the Hyperledger Fabric (HLF) platform. The approach avoided issues by eliminating central servers and their associated third-party auditors. It relied on HLF peers for the validation of transactions and record audits. Lightweight verification tags were stored on the blockchain ledger. The results showed the feasibility of managing IoT data provenance inside a Big Data framework, achieving a throughput of 600 transactions with an average response time of 500 ms, CPU utilization of 2–3% at peers and 10–20% at clients, and latency that was minimal under 1 second.

Sekhar et al., (2020) proposed experimental architecture using BDA to make common efficiency enhanced and the integration of multivariate data in smart cities excellent. The proposed architecture integrated real-time and offline data analysis systems, which enabled data processing and statistical assessment efficiently. Established authenticated datasets were used to extract the important parameters for urban operational management. Performance tests indicated the superiority of the developed platform, analyses of throughput, and processing times superior to those reported by earlier approaches. The architecture effectively demonstrated the efficacy of BDA in improving data-driven urban management.

Li and Chaomin (2020) suggested an extensive approach that brings IoT technologies together with big data tools in a unified platform for monitoring and analyzing real-time data. A Fog-assisted IoT-based Smart Real-time Healthcare Information Processing (SRHIP) system was conceptualized by transferring IoT-generated data to the Fog cloud for analytics with minimal latency. The processed data was then forwarded to a central cloud for further processing. The transmission cost, accuracy, and storage capacity constitute the measures used in determining the assessment of the Fog-assisted model. The SRHIP system reduced the transmission cost by 40.10%, completely removed compromised bytes, and decreased the size of the data to 60% of that sized by benchmark techniques.

Research gaps

- Limited focus on the practical implementation of deduplication techniques in dynamic and evolving data environments (Thanekar & Puri, 2024).
- Insufficient examination of implementation challenges in real-world hybrid systems for secure data management (Patidar et al., 2024).
- Limited exploration of the scalability, cost-effectiveness, and real-time processing capabilities of Hadoop and Spark in large-scale agricultural settings (Fatima et al., 2023).
- Limited analysis of Hyperledger Fabric's throughput in managing large-scale IoT networks within the distributed storage architecture (Demirbaga & Aujla, 2022).

3. Research Methodology

This study provides a detailed examination of the employed strategy, showing the tools utilized, methods, and how recommendations are performed as follows:

Tools Used

This section lists the tools that are employed: Python, a set of libraries like Scikit-learn, Pandas, NumPy, Matplotlib, and Seaborn for learning inside Jupyter Notebook and HDFS, a kind of distributed, fault-tolerant storage system built to support MapReduce, which, in turn, is built to support Hive for efficient big data processing.

- **Python**

Python is a cross-function, broadly applied language for data analytics, machine learning, and visualization (Python, 2021). Its rich ecosystem does include libraries like Scikit-learn, providing robust tools for building and evaluating machine learning models like regression, clustering, and classification (Aziz et al., 2021). Pandas make data manipulation and analysis exceptionally simple and powerful with its powerful DataFrame structure. NumPy for the massive multidimensional arrays and mathematical operations is essential for numerical computing (McKinney, 2022). Matplotlib and Seaborn are visualization libraries that enable the creation of detailed plots, charts, and statistical graphics to explore trends and patterns (Putri et al., 2022). SciPy addresses these with advanced mathematical functions for optimization, integration, and statistics. Jupyter Notebook is used as an interactive development environment to enact code, visualization, and narrative building all in one, well-suited for iterative workflows and report generation. Together, these packages are an integral part of Python's ecosystem of data-driven applications (Makowski et al., 2021). Figure 2 gives a visual representation of the ecosystem, which grows from foundational libraries to domain-specific projects.

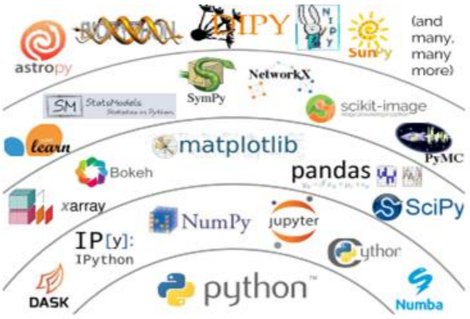

Figure 2: Schematic view of the Python scientific software ecosystem (Rodrigues et al., 2020)

- **HDFS**

HDFS is a highly scalable and fault-tolerant file storage system designed for big data applications (Kalia & Gupta, 2021). As the core component of the Hadoop ecosystem, HDFS enables the storage of large datasets across a distributed network of machines while holding high availability and reliability (Hedayati et al., 2023). It employs a master-slave architecture, with a Name Node managing metadata and directory structure and Data Nodes storing actual data (Ouatik et al., 2020). HDFS is designed for write-once, read-many applications and, therefore, is best suited for batch processing as well as analytics. The replication mechanism of HDFS ensures redundancy of data, keeping against hardware failures (Keita, 2021).

Further, HDFS supports integration with processing frameworks like MapReduce and querying tools such as Hive, enabling structured as well as unstructured data analysis (Ferraro Petrillo et al., 2021). HDFS enables distributed storage and parallel processing, which provides the foundation for efficiently dealing with extremely large data volumes in big data ecosystems (Boyko & Tkachuk, 2020). Figure 3 illustrates the visual representation of the Hadoop ecosystem.

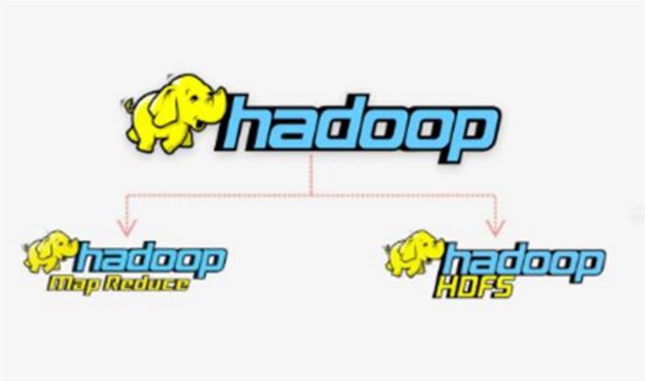

Figure 3: *Hadoop ecosystem https://www.bitwiseglobal.com/en-us/understanding-the-hadoop-adoption-roadmap/*

Proposed Methodology

IoT sensors such as DHT11/22 and HC-SR04 transmit data to platforms like ThingSpeak, Adafruit IO, or local MySQL databases. A single-node Hadoop cluster with HDFS and Hive enables distributed data storage and structured querying. Batch processing pipelines analyze data stored in HDFS using Hive queries, with results visualized or exported via Python libraries like Matplotlib and Seaborn. Machine learning models developed using Scikit-learn focus on regression for trend prediction and K-means clustering for anomaly detection, implemented via Jupyter Notebook. Performance is evaluated based on accuracy, latency, and throughput, with a comparison of local and Hadoop-based processing methods. A systematic methodology for integrating these components seamlessly is illustrated in Figure 4.

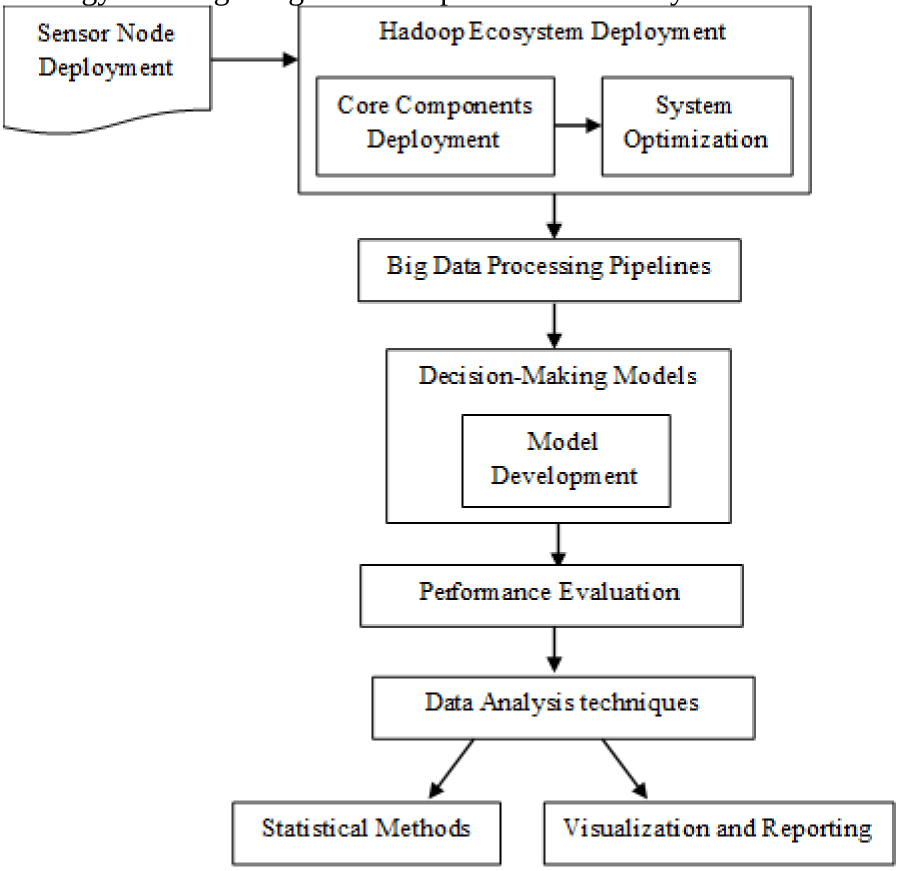

Figure 4: Proposed Methodology

Proposed Algorithm

Step 1: Sensor Node Deployment

1. **Infrastructure Configuration:**
- Let S be the set of IoT sensors, $S=\{s_1,s_2,..., s_n\}$, where s_1=DHT11/22, s_2=HC-SR04.
- Let D be the data collected from S, $D=\{d_1,d_2,...,d_k\}$.
- Data transmission:
    - If Storage=Cloud, transmit D to P_c, where P_c=ThingSpeak or Adafruit IO.
    - If Storage=Local, store D in L_s, where L_s=MySQL database or local file.
2. **Data Flow Design:**
- Define a mapping function $f: D\longrightarrow T$, where T is the target storage:
    - T=Google Drive (cloud) or T = MySQL (local).

Step 2: Hadoop Ecosystem Deployment

1. **Core Components Deployment:**
- Deploy Hadoop cluster H = {HDFS, Hive}.
- Configure HDFS for data storage:
    - HDFS(D) = D_s, where D_s is the distributed storage of D.
- Configure Hive for structured queries Q, where Q ∈ SQL queries on D_s.

Step 3: Big Data Processing Pipelines

1. **Pipeline Architecture:**
- Use a batch processing pipeline P_b:
    - Input: $D_s \rightarrow P_b$.
    - Output: Processed data D_p.
2. **Data Workflow:**
- Apply Hive queries $Q(D_s) \rightarrow D_p$.
- Export D_p to:
    - CSV file C, where $D_p \rightarrow C$, or
    - Visualize D_p using Python tools V_p, where V_p = Matplotlib/Seaborn.

Step 4: Decision-Making Models

1. **Model Development:**
- Let M be the set of machine learning models:
    - $M = \{M_1,M_2\}$, where:
        - M1 = Regression (Linear/Polynomial),
        - M2 = K-means clustering.
- Input data for M: $M(D_p)$.
2. **Implementation Scope:**
- Model execution $M(D_p) \rightarrow \{R,G\}$, where:
    - R is regression results (trend predictions),
    - G is clustering results (grouping/anomaly detection).

Step 5: Performance Evaluation Metrics

1. **Evaluation Criteria:**
- Let:

- Latency = L= Time for processing data.
- Throughput (T) = $\frac{Total\,Tasks\,Processed}{Total\ time}$

2. **Comparative Analysis:**

- Compare local processing P_l and Hadoop-based processing P_h:
  - $A(P_l)$ vs. $A(P_h)$,
  - $L(P_l)$ vs. $L(P_h)$,
  - $T(P_l)$ vs. $T(P_h)$.

Step 6: Data Analysis Techniques

1. **Statistical Methods:**

- Compute basic statistics S_t:
  - Mean $\mu = \frac{\sum_{i=1}^{N} X_i}{N}$
  - Variance $\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(d_i - \mu)^2$,

2. **Visualization and Reporting:**

- Generate plots V:
  - Line graph: $V_{line}(D_p)$,
    - Bar chart: Vbar(Dp).
- Automate reports R_a:
  - R_a = Key findings in Jupyter Notebook.

4. Results and Discussion

This section provides the outcomes of the research that are obtained after the implementation of the proposed methodology.

Evaluation Metrics

- **Minimum Delay (MinDelay)**

The shortest amount of time taken for a process to complete in the system. It represents the most efficient performance observed in the dataset (Huang et al., 2021).

$$\text{Minimum Delay} = \min(D_i) \tag{1}$$

where D_i represents the delays for all observed instances i.

- **Maximum Delay (MaxDelay)**

The longest amount of time taken for a process to complete in the system indicates the worst-case scenario (Huang et al., 2021).

$$\text{Maximum Delay} = \max(D_i) \tag{2}$$

- **Average Delay**

The mean time taken for processes provides an overall measure of the system's processing performance (Huang et al., 2021).

$$Average\ Delay = \frac{\sum_{i=1}^{N} D_i}{N} \tag{3}$$

Where N is the total number of instances.

- **Mean**

The central tendency of a dataset represents the average value of delays or other metrics (Haile et al., 2021).

$$Mean\ (\mu) = \frac{\sum_{i=1}^{N} X_i}{N} \tag{4}$$

where X_i is the value of each data point, and N is the total number of data points.

- **Variance**

A measure of the spread of data points around the mean indicates the variability or consistency of the delays (Haile et al., 2021).

$$Variance\ \sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(X_i - \mu)^2 \quad (5)$$

- **Latency**

The total time taken for a system to process and respond to a request is often measured in seconds or milliseconds (Lu & Shen, 2013).

$$Latency = End\ time - Start\ time \quad (6)$$

where End Time is the time when the task finishes and Start Time is when it begins.

- **Throughput**

The rate at which tasks or data are processed by the system is typically measured in units per second. It indicates system efficiency (Lu & Shen, 2013).

$$\text{Throughput (T)} = \frac{Total\ Tasks\ Processed}{Total\ time} \quad (7)$$

where Total Tasks Processed is the number of operations completed, and Total Time is the duration over which the tasks were processed.

Performance Evaluation

The attached Table 1 presents a detailed analysis of processed delay metrics from a Big Data processing study, focusing on the maximum, minimum, and average delays associated with specific Block IDs. The data reveals significant variance in delay times. These findings illustrate the effectiveness of delay reduction techniques, emphasizing their role in optimizing Big Data workflows for improved decision-making, as outlined in the research paper on enhancing IoT ecosystems with Big Data analytics and Hadoop frameworks.

Table 1: Processed Delay Metrics

S.No	BlockId	Processed_MaxDelay	Processed_MinDelay	Processed_AverageDelay
1	blk_-1608999687919862906	23279620000.0	20883414.0	624508500.0
2	blk_7503483334202473044	6067420000.0	103947.6	82003250.0
3	blk_-3544583377289625738	7609327.0	483231.6	1233637.0
4	blk_-9073992586687739851	1916508000.0	0.0	6456451.0
5	blk_7854771516489510256	18396530.0	2771217.6	5241321.0

The attached Figure 5 illustrates the mean values of the processed delay metrics: MaxDelay (~9.31 ns), MinDelay (~2.08 ns), and AverageDelay (~7.1 ns). The MaxDelay is significantly higher, reflecting the impact of outliers with extreme delay values. In contrast, MinDelay shows the lowest mean value, highlighting effective delay reduction across Block IDs. The AverageDelay, lying between the two, demonstrates overall consistency and optimization in delay times. These findings confirm the success of Big Data processing techniques in reducing delays, particularly minimizing extreme values, thereby enhancing the reliability and efficiency of decision-making in IoT ecosystems.

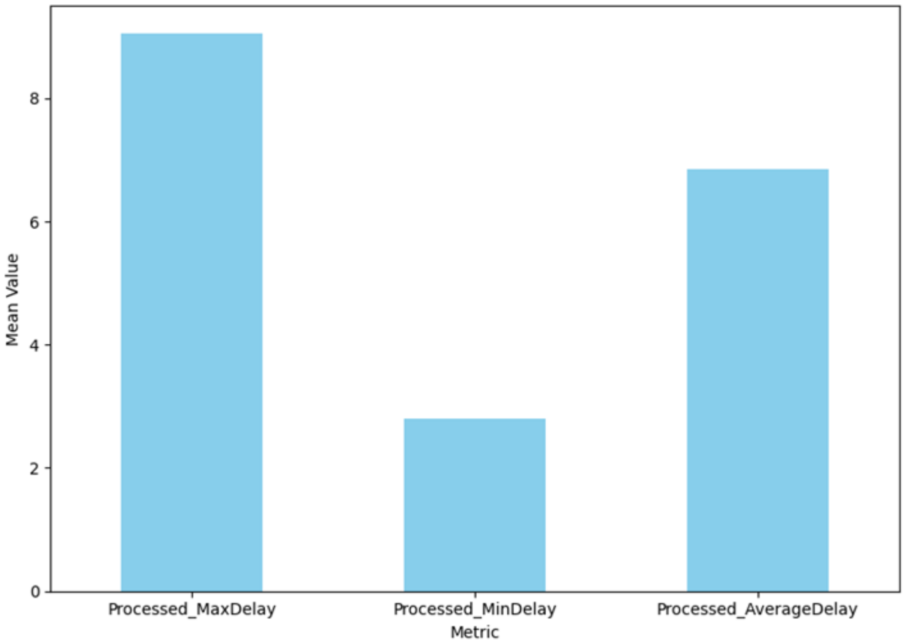

Figure 5 : Mean of Processed Delay.

Figure 6 illustrates the variance values of the processed delay metrics: MaxDelay (~5.35), MinDelay (~1.23), and AverageDelay (~4.28). The MaxDelay shows the highest variance, indicating substantial fluctuations in maximum delay values across the dataset. In contrast, MinDelay demonstrates the lowest variance, reflecting consistent reductions in minimum delays after processing. The AverageDelay exhibits moderate variance, suggesting relative stability in average delay values while still being influenced by variations in maximum delays. Overall, the graph highlights the effectiveness of Big Data processing techniques in minimizing inconsistencies, especially in minimum delays, while underscoring the need for further optimization to reduce variability in maximum delays.

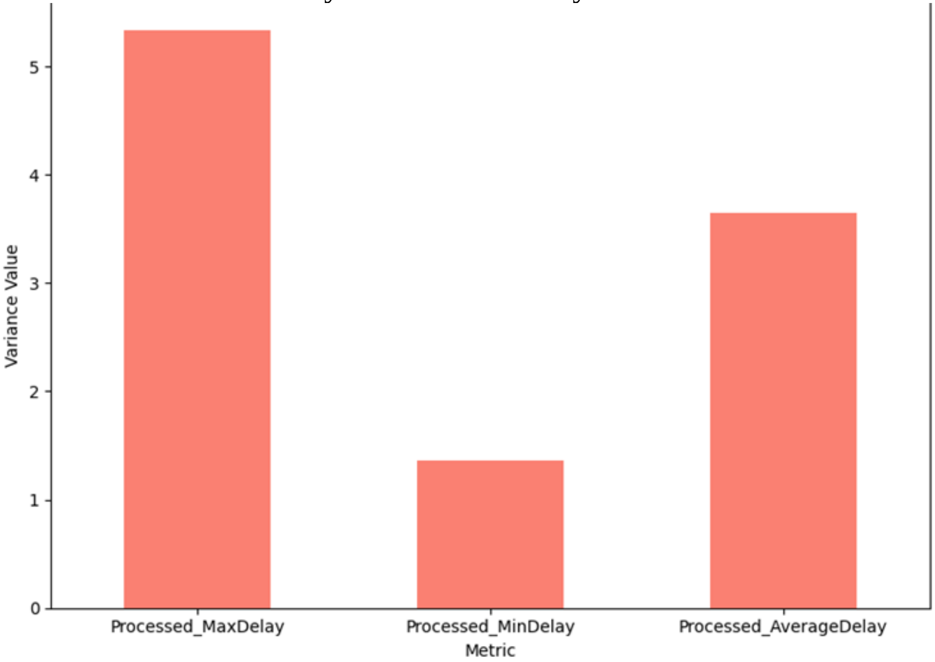

Figure 6: Variance of Processed Delays

Table 2 shows the regression predictions against the average delays for a certain Block ID, which shows performance and variability in processing efficiency. Block 1 shows the highest regression prediction (557,005,200.0) and average delay (594,770,000.0), meaning the processing delay is substantial, and there is much to be optimized. Block 2 has a moderate regression prediction (114,075,200.0) and average delay (78,098,340.0), which indicates fairly stable performance with some potential for improvement. Block 3 is characterized by a negative regression prediction of -11,072,250.0 and a minimal average delay of 1,174,892.0, suggesting possible anomalies or inconsistencies in predictive modeling. Block 4 has a regression prediction of 26,894,780.0 and an average delay of 6,149,001.0, which indicates balanced performance. Finally, Block 5 has a slight negative regression prediction of -1,691,078.0 and an average delay of 4,991,734.0, indicating relatively stable processing with minor predictive errors. These results prove that the optimization of real-time decision-making is the focus of research on IoT ecosystems. It shows through analysis of how well predictions were correlated to observed delays how effective a regression model may be and indicates where improvements could be made in it. The findings validate that enhancing predictive analytics and the inclusion of strong Big Data and Hadoop frameworks will greatly promote operational efficiency, mitigate delays, and improve decision-making systems' accuracy in an IoT environment.

Table 2: Regression Prediction

BlockId	Regression_Predictions	AverageDelay
1	557005200.0	594770000.0
2	114075200.0	78098340.0
3	-11072250.0	1174892.0
4	26894780.0	6149001.0
5	-1691078.0	4991734.0

The following table 3 presents the result of K-means clustering in anomaly detection. Here, block IDs are classified into various clusters with the aid of their delay metrics. Only Block 1 has been assigned uniquely to Cluster 1.0, where the maximum delay was at its highest (2.116329e+10), and its minimum delay is pretty high, about 17,402,845.0. It indicates a probable anomaly or outlier. All the rest of the blocks were assigned to Cluster 0.0. Block 2 has a maximum delay of 5.515836e+09 and a minimal delay of 86,623.0, showing to have been mediocre performance. Block 3 shows that the maximum delay was at 6.917570e+06 and the minimum at 402,693.0. Block 4 is at a maximum delay of 1.742280e+09 and a minimum of 0.0; this indicates variability in process times and possibly inefficient usage of processing time. Finally, Block 5 shows a maximum delay of 1.672412e+07 and a minimum delay of 2,309,348.0, which means that it is more stable than the other blocks. These clustering results align with the research's objective of enhancing decision-making in IoT ecosystems by identifying anomalies and optimizing data processing. By isolating outliers and differentiating normal performance patterns, the table supports the refinement of anomaly detection methods and validates the efficiency of the proposed Big Data and Hadoop frameworks. This facilitates targeted optimization, reducing delays, improving consistency, and ensuring robust and reliable system performance.

Table 3: K-means clustering for anomaly detection.

BlockId	Cluster_Label	MaxDelay	MinDelay
1	1.0	2.116329e+10	17402845.0
2	0.0	5.515836e+09	86623.0
3	0.0	6.917570e+06	402693.0
4	0.0	1.742280e+09	0.0
5	0.0	1.672412e+07	2309348.0

Figure 7 depicts the latency (in seconds) across various Block IDs, ranging from approximately 1 second to 4.5 seconds, with notable fluctuations. Peaks at 4.5 seconds highlight instances of high processing delays, while drops to around 1 second reflect efficient performance in certain blocks. Latency variability suggests inconsistent processing, with certain regions, such as Block IDs between 30 and 40, experiencing consistently higher delays, whereas others, like near Block IDs 10 and 20, maintain stability. These findings emphasize the need for targeted optimization in blocks with higher delays while preserving the efficiency observed in low-latency blocks to enhance overall system performance and real-time decision-making capabilities.

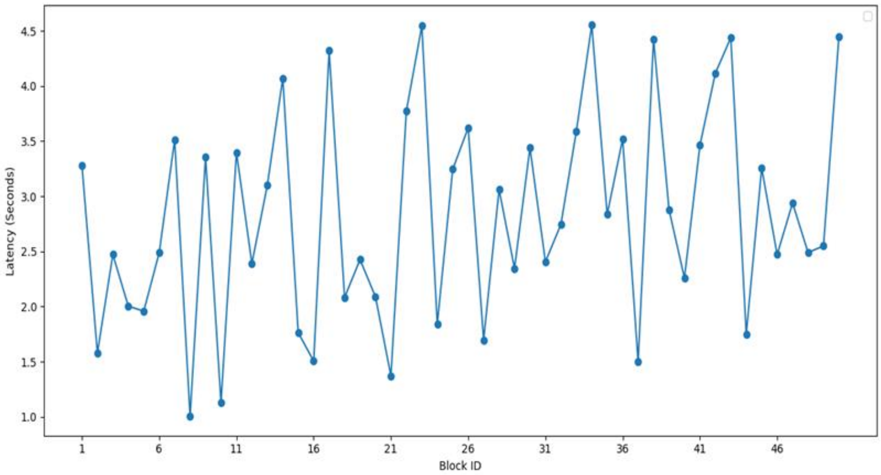

Figure 7: Latency Vs Block ID.

Figure 8 illustrates the throughput (samples per second) across various Block IDs, ranging from 0.5 samples/second to a peak of 2 samples/second, highlighting variations in processing performance. Block IDs around 6, 26, and 36 achieve the highest throughput, nearing 2 samples/second, indicating highly efficient processing, while lower throughput values, approximately 0.5 samples/second, are observed around Block IDs 11 and 41, suggesting potential inefficiencies or bottlenecks. Overall, throughput remains relatively stable across most Block IDs, fluctuating between 1–1.5 samples/second, reflecting consistent system performance with occasional dips and peaks. These results emphasize the need to optimize low-throughput blocks while leveraging the efficiency demonstrated in high-throughput areas to enhance overall system performance.

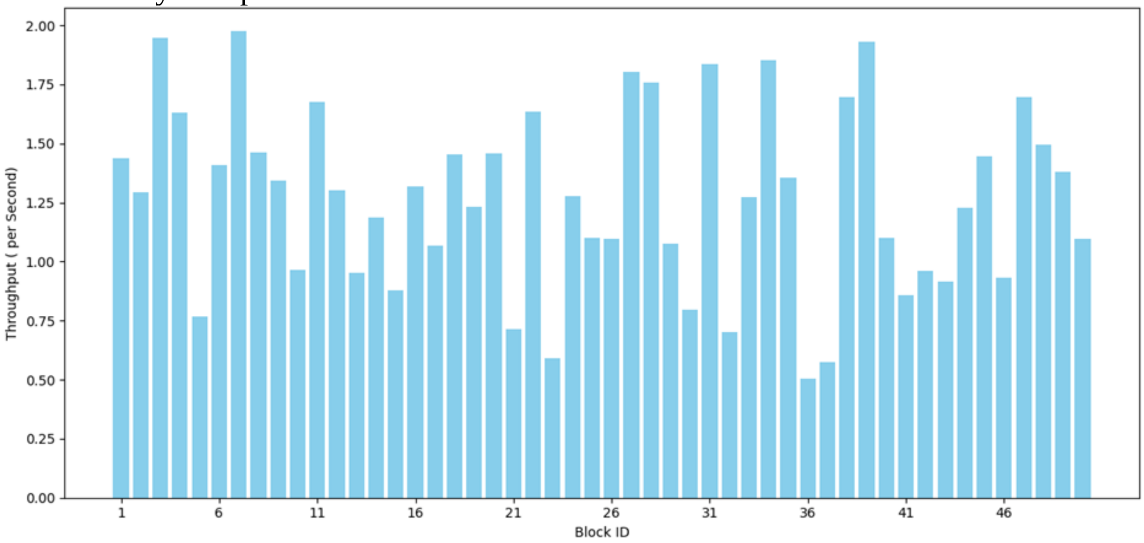

Figure 8: Throughput Vs Block ID.

5. Conclusion and Future Scope

Enhancing decision-making in IoT ecosystems involves utilizing advanced technologies to process and analyze vast amounts of data generated by interconnected devices, enabling real-time, informed decisions that improve efficiency and functionality. The research paper focuses on enhancing decision-making in IoT ecosystems using Big Data Analytics and Hadoop frameworks. As IoT devices generate massive data, traditional frameworks fail to process them in real time. The present study aims to design and evaluate an integrated model leveraging BDA and Hadoop for optimizing decision-making, improving operational efficiency, predictive maintenance, and actionable insights. The methodology includes employing Python libraries like Scikit-learn, Pandas, NumPy, Matplotlib, and Seaborn for data analysis, Hadoop Distributed File System (HDFS) for scalable storage, and Hive for structured querying. IoT sensors such as DHT11/22 and HC-SR04 transmit data to platforms like ThingSpeak, Adafruit IO, or local MySQL databases. A single-node Hadoop cluster with HDFS and Hive enables distributed data storage and structured querying. Batch processing pipelines analyze data stored in HDFS using Hive queries, with results visualized or exported via Python libraries like Matplotlib and Seaborn. Machine learning models developed using Scikit-learn focus on regression for trend prediction and K-means clustering for anomaly detection, implemented via Jupyter Notebook. Performance is evaluated based on accuracy, latency, and throughput, with a comparison of local and Hadoop-based processing methods. Evaluation parameters include latency, throughput, delay metrics, regression predictions, and K-means clustering for anomaly detection. The results reflect a tremendous reduction in delay times. The average delay is reduced to approximately 7.1 ns, the maximum delay at 9.31 ns, and the minimum delay at 2.08 ns. The throughput values vary between 0.5 and 2 samples/second, which reflects processing efficiency. Regression prediction has been done to depict processing delays. The notable prediction includes 557,005,200.0 and 114,075,200.0. K-means clustering efficiently detects anomalies. It shows that Block 1 had the maximum delay of 2.116329e+10. The current work does achieve notable improvement in real-time data processing, thus providing scalable fault-tolerant solutions to the IoT ecosystem. In the future, predictive analytics may be refined, data processing frameworks may be optimized, and hybrid systems may be researched to make data management secure.

References

1. Nathali Silva, B., Khan, M., & Han, K. (2017). Big data analytics embedded smart city architecture for performance enhancement through real-time data processing and decision-making. Wireless communications and mobile computing, 2017(1), 9429676.
2. Bibri, S. E. (2018). The IoT for smart sustainable cities of the future: An analytical framework for sensor-based big data applications for environmental sustainability. Sustainable cities and society, 38, 230-253.
3. Nisar, Q. A., Nasir, N., Jamshed, S., Naz, S., Ali, M., & Ali, S. (2021). Big data management and environmental performance: role of big data decision-making capabilities and decision-making quality. Journal of Enterprise Information Management, 34(4), 1061-1096.
4. Hu, L., & Shu, Y. (2023). Enhancing Decision-Making with Data Science in the Internet of Things Environments. International Journal of Advanced Computer Science and Applications, 14(9).

5. Hussain, F., Nauman, M., Alghuried, A., Alhudhaif, A., & Akhtar, N. (2023). Leveraging Big Data Analytics for Enhanced Clinical Decision-Making in Healthcare. IEEE Access, 11, 127817-127836.
6. Osman, A. M. S. (2019). A novel big data analytics framework for smart cities. Future Generation Computer Systems, 91, 620-633.
7. Li, C. (2020). Information processing in Internet of Things using big data analytics. Computer Communications, 160, 718-729.
8. Shahat Osman, A. M., & Elragal, A. (2021). Smart cities and big data analytics: a data-driven decision-making use case. Smart Cities, 4(1), 286-313.
9. Silva, B. N., Khan, M., Jung, C., Seo, J., Muhammad, D., Han, J., ... & Han, K. (2018). Urban planning and smart city decision management empowered by real-time data processing using big data analytics. Sensors, 18(9), 2994.
10. Arora, N., Singh, A., Shahare, V., & Datta, G. (2023). Introduction to Big Data Analytics. In Towards the Integration of IoT, Cloud and Big Data: Services, Applications and Standards (pp. 1-18). Singapore: Springer Nature Singapore.
11. Al-Jumaili, A. H. A., Muniyandi, R. C., Hasan, M. K., Paw, J. K. S., & Singh, M. J. (2023). Big data analytics using cloud computing based frameworks for power management systems: Status, constraints, and future recommendations. Sensors, 23(6), 2952.
12. Mukherjee, S., Gupta, S., Rawley, O., & Jain, S. (2022). Leveraging big data analytics in 5G-enabled IoT and industrial IoT for the development of sustainable smart cities. Transactions on Emerging Telecommunications Technologies, 33(12), e4618.
13. Vashishth, T. K., Sharma, V., Pandey, A., & Tomer, T. Innovative Advancements in Big Data Analytics: Navigating Future Trends and Direction with Hadoop Integration.
14. Demertzis, K., Rantos, K., & Drosatos, G. (2020). A dynamic intelligent policies analysis mechanism for personal data processing in the IoT ecosystem. Big Data and Cognitive Computing, 4(2), 9.
15. Allam, S. (2017). Exploratory Study for Big Data Visualization in the Internet of Things. Sudhir Allam," EXPLORATORY STUDY FOR BIG DATA VISUALIZATION IN THE INTERNET OF THINGS", International Journal of Creative Research Thoughts (IJCRT), ISSN, 2320-2882.
16. Zineb, E. F., Najat, R. A. F. A. L. I. A., & Jaafar, A. B. O. U. C. H. A. B. A. K. A. (2021). An intelligent approach for data analysis and decision making in big data: a case study on e-commerce industry. International Journal of Advanced Computer Science and Applications, 12(7).
17. Koren, O., Hallin, C. A., Perel, N., & Bendet, D. (2019). Decision-making enhancement in a big data environment: application of the k-means algorithm to mixed data. Journal of Artificial Intelligence and Soft Computing Research, 9(4), 293-302.
18. Liu, Y., He, K., & Qin, F. (2021). Remote Sensing Big Data Analysis of the Lower Yellow River Ecological Environment Based on Internet of Things. Journal of Sensors, 2021(1), 1059517.
19. Paramesha, M., Rane, N. L., & Rane, J. (2024). Big data analytics, artificial intelligence, machine learning, internet of things, and blockchain for enhanced business intelligence. Partners Universal Multidisciplinary Research Journal, 1(2), 110-133.
20. Rehman, A., Naz, S., & Razzak, I. (2022). Leveraging big data analytics in healthcare enhancement: trends, challenges, and opportunities. Multimedia Systems, 28(4), 1339-1371.

21. Rossi, L., & Bianchi, G. (2022). Big Data Analytics: Harnessing the Power of Data Science for Enhanced Decision-Making in Modern Business Environments. MZ Computing Journal, 3(2).
22. Ibrahim, F., & Aoun, M. (2022). Improving query efficiency in heterogeneous big data environments through advanced query processing techniques. Journal of Contemporary Healthcare Analytics, 6(6), 40-64.
23. Desarkar, A., & Das, A. (2017). Big-data analytics, machine learning algorithms, and scalable/parallel/distributed algorithms. Internet of Things and big data technologies for next generation healthcare, 159-197.
24. Nguyen, V. Q., Nguyen, V. H., Nguyen, M. Q., Huynh, Q. T., & Kim, K. (2022, December). Big Data Knowledge Acquisition Platform for Smart Farming. In Proceedings of the 11th International Symposium on Information and Communication Technology (pp. 390-397).
25. Ikegwu, A. C., Nweke, H. F., Anikwe, C. V., Alo, U. R., & Okonkwo, O. R. (2022). Big data analytics for data-driven industry: a review of data sources, tools, challenges, solutions, and research directions. Cluster Computing, 25(5), 3343-3387.
26. Sharma, P., & Barua, S. (2023). From data breach to data shield: the crucial role of big data analytics in modern cybersecurity strategies. International Journal of Information and Cybersecurity, 7(9), 31-59.
27. Palanisamy, V., & Thirunavukarasu, R. (2019). Implications of big data analytics in developing healthcare frameworks–A review. Journal of King Saud University-Computer and Information Sciences, 31(4), 415-425.
28. Haddad, O., Fkih, F., & Omri, M. N. (2024). An intelligent sentiment prediction approach in social networks based on batch and streaming big data analytics using deep learning. Social Network Analysis and Mining, 14(1), 150.
29. Hasanpuri, V., & Diwaker, C. (2024). Original Research Article An enhanced distributed framework for real-time performance testing of large scale IoT dataset using big data analytic tools. Journal of Autonomous Intelligence, 7(1).
30. Thanekar, S. A., & Puri, G. D. (2024). Improved Job Execution in Hadoop using the Task Deduplication Approach. Educational Administration: Theory and Practice, 30(5), 13392-13403.
31. Patidar, N., Zreiqat, S., Mahesh, S., & Woo, J. (2024). Cyberattack Data Analysis in IoT Environments using Big Data. arXiv preprint arXiv:2406.10302.
32. Rahmani, A. M., Chamzini, E. Y., pourshaban, M., & Hosseinzadeh, M. (2024). Scheduling of Big Data Workflows in the Hadoop Framework with Heterogeneous Computing Cluster. Arabian Journal for Science and Engineering, 1-13.
33. Fatima, S. A., Nasim, S. F., & Ahmed, S. (2023). Enhancing Agricultural Operations: Big Data Analytics Using Distributed and Parallel Computing. International Journal of Emerging Engineering and Technology, 2(2), 1-7.
34. Demirbaga, U., & Aujla, G. S. (2022). MapChain: A blockchain-based verifiable healthcare service management in IoT-based big data ecosystem. IEEE Transactions on Network and Service Management, 19(4), 3896-3907.
35. Honar Pajooh, H., Rashid, M. A., Alam, F., & Demidenko, S. (2021). IoT Big Data provenance scheme using blockchain on Hadoop ecosystem. Journal of Big Data, 8, 1-26.
36. Sekhar, J. C., & Pratap, V. K. (2020). Design and Implementation of Smart City Big Data Processing Platform Using Big Data Analytics for Decision Management System. Mathematical Statistician and Engineering Applications, 69(1), 617-627.

37. Li, C. (2020). Information processing in Internet of Things using big data analytics. Computer Communications, 160, 718-729.
38. Python, W. (2021). Python. Python releases for windows, 24.
39. Aziz, Z. A., Abdulqader, D. N., Sallow, A. B., & Omer, H. K. (2021). Python parallel processing and multiprocessing: A rivew. Academic Journal of Nawroz University, 10(3), 345-354.
40. McKinney, W. (2022). Python for data analysis. " O'Reilly Media, Inc.".
41. Putri, G. H., Anders, S., Pyl, P. T., Pimanda, J. E., & Zanini, F. (2022). Analyzing high-throughput sequencing data in Python with HTSeq 2.0. Bioinformatics, 38(10), 2943-2945.
42. Makowski, D., Pham, T., Lau, Z. J., Brammer, J. C., Lespinasse, F., Pham, H., ... & Chen, S. A. (2021). NeuroKit2: A Python toolbox for neurophysiological signal processing. Behavior research methods, 1-8.
43. Rodrigues, E., Krikler, B., Burr, C., Smirnov, D., Dembinski, H., Schreiner, H., ... & Das, P. (2020). The Scikit HEP Project overview and prospects. In EPJ Web of Conferences (Vol. 245, p. 06028). EDP Sciences.
44. Kalia, K., & Gupta, N. (2021). Analysis of hadoop MapReduce scheduling in heterogeneous environment. Ain Shams Engineering Journal, 12(1), 1101-1110.
45. Hedayati, S., Maleki, N., Olsson, T., Ahlgren, F., Seyednezhad, M., & Berahmand, K. (2023). MapReduce scheduling algorithms in Hadoop: a systematic study. Journal of Cloud Computing, 12(1), 143.
46. Ouatik, F., Erritali, M., & Jourhmane, M. (2020). Student orientation using machine learning under MapReduce with Hadoop. J. Ubiquitous Syst. Pervasive Networks, 13(1), 21-26.
47. Keita, M. (2021). Big Data et Technologies de Stockage et de Traitement des Données Massives: Comprendre les bases de l'écosystème HADOOP (HDFS, MAPREDUCE, YARN, HIVE, HBASE, KAFKA et SPARK).
48. Ferraro Petrillo, U., Palini, F., Cattaneo, G., & Giancarlo, R. (2021). FASTA/Q data compressors for MapReduce-Hadoop genomics: space and time savings made easy. BMC bioinformatics, 22, 1-21.
49. Boyko, N., & Tkachuk, N. (2020, November). Processing of Medical Different Types of Data Using Hadoop and Java MapReduce. In IDDM (pp. 405-414).
50. https://www.bitwiseglobal.com/en-us/understanding-the-hadoop-adoption-roadmap/
51. Huang, C. Q., Zheng, C. B., Yang, F., & Su, C. Y. (2021). Performance assessment of multivariate process using time delay matrix. Journal of Process Control, 98, 10-17.
52. Haile, H., Grinnemo, K. J., Ferlin, S., Hurtig, P., & Brunstrom, A. (2021). End-to-end congestion control approaches for high throughput and low delay in 4G/5G cellular networks. Computer Networks, 186, 107692.
53. Lu, N., & Shen, X. S. (2013). Scaling laws for throughput capacity and delay in wireless networks—A survey. IEEE Communications Surveys & Tutorials, 16(2), 642-657.