International Journal of Computer Sciences and Engineering
Vol.**11**, Special Issue.**1**, pp.**270-276**, November **2023**
ISSN: 2347-2693 (Online)
Available online at: www.ijcseonline.org

ISSN: 2347-2693 (E)

Review Paper

A Review on Big Data Architecture and It's Application for Future Aspect

Srinjoy Saha[1*], Sneha Nej[2], Rupa Saha[3], Debmitra Ghosh[4], Anal Rauth[5]

[1,2,3]Computer Application, Narula Institute of Technology, Kolkata, India
[4,5]Computer Science and Engineering, JIS University, Kolkata, India

**Corresponding Author: 6srinjoysaha6@gmail.com*

Abstract: This increase in data is not going to slow down anytime soon. Data too large or complex for traditional database systems can be ingested, processed, and analyzed using a big data architecture. Depending on the capabilities of a company's users and tools, the threshold for entering into the big data realm may differ. This paper is mainly focuses on essential questions for Big Data Architecture – What is Big Data Architecture, Big Data Architecture Layers, Types of Big Data Architecture, Big Data Architecture Application, Benefits of Big Data Architecture and Big Data Architecture Challenges.

Keywords: Big Data Architecture, Data Lake, Lambda Architecture, Kappa Architecture, Data Ingestion, Data Visualization.

1. Introduction

Big Data architecture is crucial due to the escalating volume, velocity, and variety of data generated in today's digital landscape. It provides a structured framework to efficiently handle and extract meaningful insights from these massive datasets.

Firstly, Big Data architecture enables organizations to process and analyze vast amounts of information that traditional systems struggle to manage. By employing distributed computing techniques, such as Hadoop and Spark, it can handle data at scale, allowing for deeper analysis and pattern recognition.

Secondly, it supports real-time or near-real-time data processing, enabling timely decision-making. Businesses can respond swiftly to market changes, customer behaviors, and emerging trends, gaining a competitive edge.

Thirdly, proper architecture ensures data quality, integration, and security. It facilitates the organization, storage, and retrieval of data from disparate sources, contributing to accurate analysis and informed choices. Additionally, robust security measures safeguard sensitive information in compliance with regulations.

Furthermore, Big Data architecture empowers data-driven innovation. It facilitates the development of advanced analytics, machine learning models, and predictive algorithms that unveil valuable insights, optimize processes, and drive innovation.
In essence, Big Data architecture is pivotal as it transforms raw data into actionable intelligence, enabling businesses to make informed decisions, improve operational efficiency, and create innovative products and services in our data-driven world. Systems and software used for managing Big Data are referred to as Big Data architecture; they must be able to handle their scale, complexity, and variety. An organization's big data architecture answers the question of how to deal with a large amount of data and consists of plans for solutions based on the demands of the organization. As well as allowing users to access and analyze data differently, it must also be flexible enough to support the needs of different users. Data management processes and organizational structures are part of the Big Data pipeline architecture, which is necessary for users to be able to effectively work with Big Data.[1] In Big Data, there are several different types of architecture, including Azure Big Data architecture, Hadoop architecture, and Spark architecture.

2. Big Data Architecture Layers

A more detailed representation of a Big Data architecture can be divided into several layers, each serving a specific purpose in handling and processing large volumes of data. Here are the four main layers in a typical Big Data architecture:

1. Data Sources Layer: This is where data originates. It includes various sources such as sensors, logs, social media, databases, and more. In a mathematical representation, you can denote it as DS = {D₁, D₂, ..., D_n}, where n is the number of data sources.
2. Data Ingestion and Collection Layer: This layer handles the process of collecting, transporting, and ingesting data from various sources into the Big Data ecosystem. It often involves technologies like Apache Kafka, Apache NiFi, or other data ingestion frameworks. In mathematical terms, you could represent this layer as I(D_i) = Ingested Data, where I is the ingestion function.

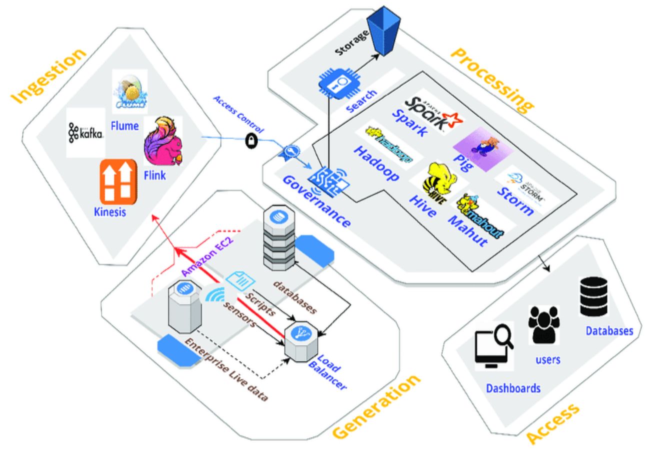

Figure 1: Diagram of Big Data Sources and Storage Layer

3. Data Storage and Processing Layer: This is a critical layer where data is stored and processed. It usually includes both batch processing and real-time/streaming processing. Technologies like Hadoop Distributed File System (HDFS), Apache Spark, and Apache Flink are commonly used here. Mathematically, you might represent it as S(Ingested Data) = Stored Data for storage and P(Stored Data) = Processed Data for processing.

4. Data Analysis and Visualization Layer: In this layer, the processed data is analyzed to extract insights and make decisions. It involves various tools and frameworks like Apache Hive, Apache Pig, machine learning libraries, and data visualization tools. The mathematical representation could be A(Processed Data) = Analysis Result.

Combining these layers, you could represent the flow of data through the Big Data architecture in mathematical terms like this:

DS = {D₁, D₂, ..., D_n}(Data Sources)
I(D_i) = Ingested Data (Ingestion)
S(Ingested Data) = Stored Data (Storage)
P(Stored Data) = Processed Data (Processing)
A(Processed Data) = Analysis Result (Analysis) [2][3]

3. Types of Big Data Architecture

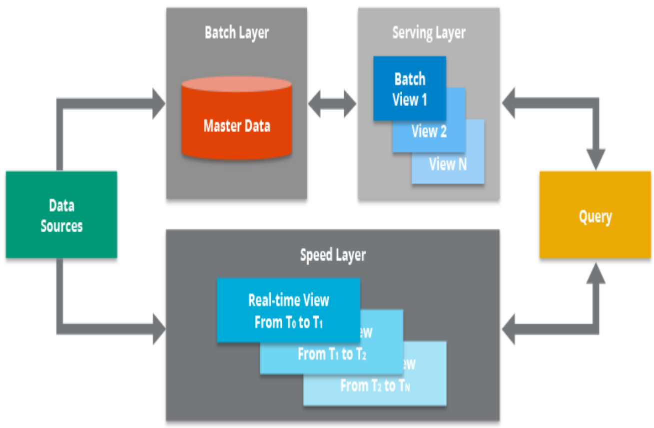

Figure 2: Diagram of Lambda Architecture

3.1) Lambda Architecture: Using Lambda architecture, one can solve the problem of computing arbitrary functions using batch data (static) as well as real-time data (processing). A deployment model with this approach reduces latency and preserves accuracy while preserving minimal errors.

The Lambda Architecture is a data processing architecture designed to handle massive amounts of data in real-time, combining batch processing, stream processing, and serving layers to provide a comprehensive solution for big data processing. It was introduced by Nathan Marz in his book "Big Data: Principles and best practices of scalable real-time data systems."

The Lambda Architecture is designed to address challenges in processing and analyzing large volumes of data in situations where traditional batch processing might not provide the required speed and responsiveness. It's particularly useful in scenarios where data needs to be ingested, processed, and analyzed in real-time or near real-time.

The architecture consists of three main layers:

1. Batch Layer: The batch layer is responsible for processing and storing large volumes of data over an extended period. This layer performs complex computations, including data transformation, cleansing, aggregation, and building of precomputed views. The processed results are stored in a batch layer storage system, often a distributed file system like Hadoop HDFS.

2. Speed Layer: The speed layer handles real-time data processing. It processes incoming data streams in near real-time and generates incremental updates to the batch layer's precomputed views. This layer typically uses stream processing frameworks like Apache Kafka, Apache Flink, or Apache Storm.

3. Serving Layer: The serving layer provides a way to query and retrieve the results from the processed data. It combines the batch layer's precomputed views and the real-time updates from the speed layer. This layer allows users to interact with the data and get both historical and up-to-the-moment insights.

By combining these three layers, the Lambda Architecture aims to provide a balanced solution for handling both batch and real-time data processing requirements. However, it also introduces complexities due to the need to manage both batch and real-time processing, handle data consistency between layers, and maintain and manage different processing engines. One of the challenges with the Lambda Architecture is managing the complexity of maintaining and updating code and configurations for both batch and real-time processing components. To address this, newer architectures like the Kappa Architecture have emerged, which focuses solely on stream processing to simplify the overall system.

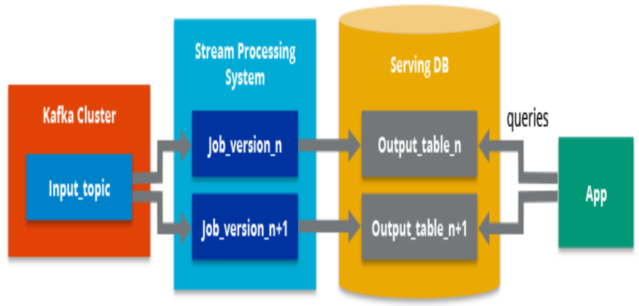

Figure 3: Diagram of Kappa Architecture

3.2) Kappa Architecture: Lambda architecture is primarily used to process real-time streaming data. Kappa architecture, on the other hand, is used to process batch data. Furthermore, Kappa replaces the data sourcing medium with message queues, which reduces the additional costs associated with Lambda.

The Kappa Architecture is a data processing framework designed for handling real-time data streams. It is an alternative to the more well-known Lambda Architecture, which is used for batch and stream processing. The Kappa Architecture simplifies the overall architecture by focusing solely on real-time data processing, reducing the complexity that can arise from managing both batch and stream processing pipelines in the Lambda Architecture.

Components of the Kappa Architecture:
1. *Ingestion Layer:* The data ingestion layer collects real-time data from various sources and feeds it into the processing pipeline. This could include sources like IoT devices, sensors, logs, and other data streams.

2. *Stream Processing:* The core of the Kappa Architecture is the stream processing layer. It processes incoming data streams in real time, applying transformations, aggregations, and other operations as needed. Apache Kafka is often used as a central component for managing the stream of data.

3. *Storage Layer*: The Kappa Architecture typically uses a log-centric storage system, like a distributed append-only log. This storage layer stores both raw incoming data and the processed results. Instead of having separate batch and serving layers like in Lambda Architecture, Kappa Architecture combines them into a single layer.

4. *Serving Layer:* In Kappa Architecture, the serving layer directly serves the processed data to consumers, whether they are end-users, applications, or analytics tools. This simplifies the architecture by eliminating the need for complex batch processing and synchronization between batch and serving layers.
[4][5]

4. Mathematical Representation

Here's a simplified mathematical representation of a Big Data architecture using mathematical symbols:

Big Data Architecture = {Data Sources, Data Ingestion, Data Storage, Data Processing, Data Analysis}
Data Sources = {D1, D2, ..., Dn}
Data Ingestion = Σ(Ingestion_i), where i = 1 to n
Data Storage = ∪(Storage_j), where j = 1 to m
Data Processing = ⊕(Processing_k), where k = 1 to p
Data Analysis = α(Analysis_l), where l = 1 to q

In this representation:
- D1, D2, ..., Dn represent different data sources.
- Ingestion_i represents the ingestion process for each data source.
- Storage_j represents different storage components.
- Processing_k represents various data processing operations.
- Analysis_l represents different types of data analysis.

Please note that this representation is quite abstract and simplified, as actual Big Data architectures can be more complex and involve a variety of technologies and components.

5. Data Lake

There is no hierarchy or organization in a data lake, only unstructured data. An organization can store all of its data on a Data Lake in order to process it later through democratization of data. Data stored on them can come from any source and can be treated as generally as possible. The Data Lake has a flat architecture, which differs from a hierarchical Data Warehouse.

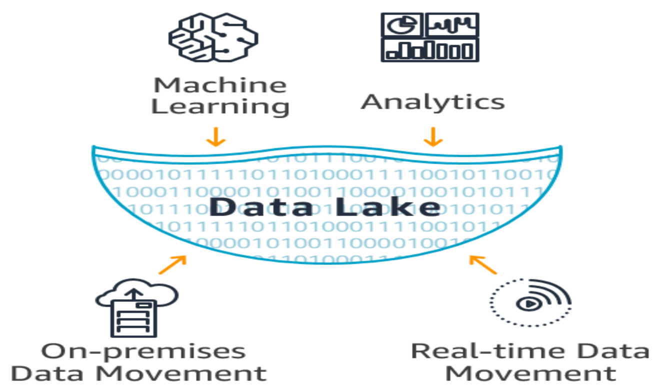

Figure 4: Diagram of Data Lake

A data lake is a centralized repository that stores vast amounts of raw, unprocessed data in various formats, such as structured, semi-structured, and unstructured data. Unlike traditional data warehouses that require predefined schemas, data lakes use a "schema-on-read" approach. This means that data is stored in its raw format, and the structure is applied only when the data is retrieved or analyzed. This flexibility makes data lakes particularly useful for handling diverse and evolving data sources.

Data lakes are designed to accommodate the needs of modern data-intensive applications, especially in the realm of Big Data and advanced analytics. They enable organizations to collect, store, and analyze massive volumes of data efficiently. Data lakes can store data from sources like IoT devices, social media platforms, logs, and more. This flexibility helps organizations harness the full potential of their data by enabling in-depth analysis and insights.

One of the key advantages of data lakes is their scalability. They can scale horizontally by adding more storage nodes or clusters as data volume grows. This makes data lakes well-suited for handling the ever-increasing amounts of data generated by organizations.

Data lakes also support a wide range of analytics use cases, including data exploration, batch processing, real-time analytics, and machine learning. With the schema-on-read approach, data scientists and analysts can work with the raw data directly, allowing for more agile and exploratory analysis.

However, data lakes also come with challenges. The lack of predefined structure can lead to issues related to data quality and consistency. Effective data governance is crucial to ensure that data in the lake is accurate, reliable, and compliant with regulations. Additionally, managing access controls and security is vital to protect sensitive information.

In summary, a data lake is a versatile and scalable solution for storing and managing large volumes of diverse data. It empowers organizations to unlock insights from their data, supporting various analytics tasks while requiring careful planning and management to ensure data quality and security.
The raw data in data lakes is typically stored in its native format and stores a tremendous amount of data. Identifiers and metadata information are assigned to each data element in an Enterprise Data Lake.[6][7] Data lakes provide on-demand data: based on search criteria, a subset of data can be queried for analysis and displayed as needed.

6. Experiment and Result

Here's an example of a Big Data architecture experiment along with hypothetical results:
Experiment: Analyzing Customer Behavior for E-commerce Personalization.
Objective: To analyze customer behavior data from an e-commerce website to improve personalization and recommendation systems.

Architecture Components:
Data Collection: Collect customer interaction data, including clicks, purchases, product views, and time spent on pages.
Data Storage: Store the collected data in a distributed storage system (e.g., Hadoop HDFS or Amazon S3).
Data Processing: Use Apache Spark for data processing and analysis.
Machine Learning: Develop recommendation models using machine learning algorithms (e.g., collaborative filtering) to provide personalized product recommendations.
Dashboard and Visualization: Visualize the results using tools like Tableau or Power BI.

Hypothetical Results:

Customer ID	Total Clicks	Total Purchases	Average Time on Site (minutes)	Recommended Products
001	120	5	8	Product A, Product C, Product F
002	50	2	12	Product B, Product E, Product H
003	220	8	6	Product C, Product D, Product G
004	75	3	10	Product A, Product F, Product I
005	90	4	7	Product B, Product E, Product H

Figure 5: Outcome of the experiment

Observations:
Customers with higher average time spent on the site tend to have more purchases.
Products A and B are frequently recommended, indicating their popularity.
Customer preferences differ; some prefer more popular products (C, D, F), while others prefer specific ones (E, H, I).
There's potential to tailor marketing strategies based on individual preferences and behaviors.
This example demonstrates how Big Data architecture can be used to analyze customer behavior and provide personalized recommendations, leading to insights that can drive marketing and sales strategies. The results table summarizes customer data and their corresponding behaviors, aiding decision-making for improving customer experiences.

7. Big Data Architecture Application

Utilizing and implementing big data applications is an important aspect of a big data architecture. Specifically, the following big data applications are used and applied:

- *Business Analytics and Intelligence:* Big Data architecture helps organizations analyze customer behavior, market trends, and operational efficiency to make informed business decisions.

- *Healthcare and Life Sciences:* It aids in genomic research, personalized medicine, and disease prediction through large-scale data analysis.

- *Finance:* Big Data architecture is used for fraud detection, risk assessment, algorithmic trading, and customer sentiment analysis.

- *Retail:* It enables personalized marketing, inventory management, demand forecasting, and customer experience improvement.

- *Telecommunications:* Helps in analyzing call records, network performance, and customer preferences for service optimization.
- *Energy and Utilities:* Big Data architecture assists in smart grid management, energy consumption analysis, and predictive maintenance of assets.
- *Transportation and Logistics:* Used for route optimization, fleet management, and real-time tracking. [8][9]

8. Benefits of Big Data Architecture

Process and store massive amounts of data is one of the primary advantages of Big Data architectures. Unstructured, semi-structured, or structured data sets are considered Big Data, by definition. A traditional database management tool is ineffective at managing such voluminous and complex data sets. Scientific, engineering, medical, and business analytics fields are especially suited to big data architectures.

It might be necessary to process millions of images for specific anomalies or characteristics of a satellite or robotic vehicle in the field of science and engineering, for example. Many thousands of patients might be subjected to genetic tests in medicine to determine if a specific gene makes them susceptible to a particular illness. To understand how users feel about your brand or business, you might analyze millions of social media feeds. Data ingesting, storing, and processing are done as quickly as possible with Big Data architectures by integrating multiple technologies.

Big Data architecture offers a wide range of benefits for organizations dealing with massive volumes of data. Here are some key advantages:
1. Scalability: Big Data architecture allows organizations to scale their infrastructure easily as data volume and processing demands increase. This scalability ensures that the system can handle large amounts of data without compromising performance.

2. Improved Decision-Making: By processing and analyzing large datasets, organizations can gain valuable insights and make informed decisions. Big Data architecture enables advanced analytics, predictive modeling, and data-driven strategies that can lead to competitive advantages.

3. Real-time Analytics: Big Data architecture supports real-time or near-real-time processing of data streams. This enables organizations to respond quickly to changing conditions, detect anomalies, and capitalize on time-sensitive opportunities.

4. Cost Efficiency: With Big Data architecture, organizations can choose between various storage and processing technologies based on their needs. This flexibility helps optimize costs by selecting the most appropriate tools for different stages of data processing.

5. Data Variety: Big Data architecture can handle diverse types of data, including structured, semi-structured, and unstructured data. This is crucial as organizations often deal with data from various sources, such as text, images, videos, social media, sensor data, and more.

6. Enhanced Data Storage: Big Data architecture offers distributed storage solutions that can accommodate vast amounts of data across multiple nodes. This not only ensures data redundancy and fault tolerance but also reduces the risk of data loss.

7. Parallel Processing: Distributed processing frameworks, such as Hadoop and Spark, allow for parallel processing of data across multiple nodes, significantly speeding up data processing tasks. This parallelism enables faster insights and quicker decision-making.

8. Data Integration: Big Data architecture facilitates the integration of data from different sources and systems. This helps create a holistic view of the data and can lead to insights that would be difficult to obtain from isolated data silos.

9. Predictive Analytics: By analyzing historical data and patterns, organizations can build predictive models that forecast future trends and outcomes. This can guide strategic planning and proactive decision-making.

10. Personalization and Customer Insights: Big Data architecture enables organizations to gather and analyze customer data, providing insights into customer behavior, preferences, and needs. This information can be used to tailor products, services, and marketing strategies to individual customers.

11. Innovation and Research: Researchers can utilize Big Data architecture to process and analyze large datasets for scientific research, medical studies, climate modeling, and more. This can lead to breakthroughs and discoveries that were previously unattainable.

12. Competitive Advantage: Organizations that effectively harness Big Data can gain a competitive edge by identifying market trends, optimizing operations, improving customer experiences, and developing innovative products and services. Overall, Big Data architecture empowers organizations to make sense of vast and complex datasets, transforming raw information into actionable insights that drive strategic decisions, innovation, and growth. [10]

9. Big Data Architecture Challenges

As a frontier technology, Big Data architecture has some potential pitfalls, but if built correctly, it can save money and help predict important trends. There is often a cost associated with adopting a Big Data architecture that holds back a Big Data project. There is a wide range of budget requirements for Big Data applications depending on their architecture, their components and tools, their management and maintenance, and whether they are built in-house or outsourced.

Using a detailed analysis of their needs and a budget that reflects those needs can help companies overcome this challenge. Companies have to sort out and prepare data for further analysis using other data types when they receive information from different sources and ensure consistency of the data formats.

Big Data architecture, while offering immense benefits, presents several challenges due to the sheer volume, variety, and velocity of data. Handling diverse data types, such as structured, semi-structured, and unstructured data from various sources, poses integration complexities. The enormous data volume strains storage and processing resources, demanding scalable infrastructure and optimized resource allocation. Real-time processing requirements for data streams introduce the challenge of maintaining accuracy and reliability while managing high data velocities.

Ensuring data quality and accuracy, or veracity, is hindered by potential inaccuracies and inconsistencies within large datasets. Data privacy and security concerns escalate with the storage and processing of sensitive information, necessitating robust security measures to prevent breaches. Selecting appropriate tools from a wide array of options and finding skilled personnel proficient in specialized Big Data technologies pose hurdles in system setup and management.

Cost management becomes challenging as data volume grows, impacting storage, processing, and infrastructure expenses. The distributed nature of Big Data architecture requires understanding of distributed computing concepts, making setup and maintenance complex. Regulatory compliance and data governance, especially in environments subject to regulations like GDPR or HIPAA, become intricate in the context of vast data repositories.

Latency and performance considerations arise in real-time analytics, necessitating high-performance and fault-tolerant designs. Organizations must manage change effectively, as transitioning to Big Data architecture may require process overhauls and workforce training. In sum, addressing these challenges mandates meticulous planning, technological expertise, and adaptable strategies to harness the full potential of Big Data architecture. [11]

10. Conclusion and Future Scope

With the growing collection of and storage of large amounts of data, the term "Big Data" has become increasingly popular. Large data volumes can be handled by big data architectures because traditional database systems cannot handle these volumes. Big Data is often described as data sets that contain enormous volumes, velocities, and varieties, but there is no single definition for the term. Having large amounts of data to process, ingest, and analyze is crucial to the success of big data infrastructure and solutions. The big data architecture framework serves as a reference blueprint for big data infrastructures and solutions. Each organization has specific needs and goals, so the type of Big Data architecture that is best suited to them will vary.

Big Data has transformed industries by enabling organizations to harness the power of data. It has facilitated better decision-making, improved operational efficiency, and the creation of new business models. However, challenges such as data privacy, security, and the need for skilled professionals to manage and interpret the data remain significant.

Future Scope:

The future of Big Data holds several exciting possibilities:

1. *Advanced Analytics:* Continued development of AI and machine learning techniques will allow organizations to extract deeper insights and predictions from Big Data.
2. *Edge Computing:* Processing data closer to its source (IoT devices, sensors) will reduce latency and enhance real-time analytics capabilities.
3. *Data Privacy and Ethics:* As data regulations evolve, there will be a greater emphasis on ensuring data privacy and ethical data usage.
4. *Hybrid Cloud Architectures:* Organizations will adopt hybrid cloud setups to balance the advantages of on-premises and cloud-based data storage and processing.
5. *Graph Analytics:* Graph databases and analytics will gain importance for understanding complex relationships in data, useful for social networks, fraud detection, and recommendation systems.
6. *Automated Data Management:* AI-powered tools will manage and curate data, reducing the manual effort required for data cleaning and preparation.
7. *Data Monetization:* Organizations will explore ways to monetize their data by sharing insights with other companies or creating new data-driven products and services.

In conclusion, Big Data architecture has enabled a wide range of applications across industries, leading to data-driven decision-making and innovation. The future of Big Data will continue to evolve with advancements in technology, regulations, and the growing need for efficient data utilization. [12]

Authors' Contributions:

The authors confirm contribution to the paper as follows: study conception: Srinjoy Saha, draft manuscript preparation: Sneha Nej, data collection: Rupa Saha, data analysis: Debmitra Ghosh and interpretation of results: Anal Rauth. All authors reviewed the results and approved the final version of the manuscript.

Acknowledgments

We would like to express our sincere gratitude to all those who supported and contributed to the completion of this research project. Without their invaluable assistance, guidance, and encouragement, this work would not have been possible.

First and foremost, we extend our deepest appreciation to Ms. Debrupa Pal, Assistant Professor of Narula Institute of Technology, for her continuous mentorship, insightful feedback, and unwavering support throughout the duration of this research. Her expertise and guidance played a pivotal role in shaping the direction of this study. Apart from that, this research work is not funded by any organization or any person.

We are also thankful to the members of the computer application department for their valuable discussions, critical insights, and collaboration that enriched our understanding and perspective on the subject matter.

Furthermore, we would like to thank our friends and family for their unwavering encouragement and understanding during the often-demanding phases of this project. Their emotional support provided the necessary foundation for our perseverance.

Lastly, we express our appreciation to the participants who willingly volunteered their time and insights for our study. Their contributions were invaluable in shaping the outcomes and conclusions of this research.

Funding Source

The research paper was conducted without any external funding support. All costs and resources required for the study were borne by the authors themselves. Despite the absence of a dedicated funding source, the research was carried out with diligence and dedication, utilizing existing institutional facilities and personal contributions. This self-funded approach highlights the authors' commitment to advancing knowledge and contributing to their field of study, even in the absence of financial backing. The research outcome underscores the significance of individual dedication and passion in driving scientific inquiry and progress.

References

[1]. Barton, D., & Court, D. (2012). Making Advanced Analytics Work For You. Harvard Business Review, 90(10), pp.79–83, 2012.
[2]. McAfee, A., & Brynjolfsson, E. (2012). Big Data: The Management Revolution. Harvard Business Review, 90(10), pp.60–68, 2012.
[3]. Wu, X., Wu, G., & Ding, W. (2014). Data Mining with Big Data. IEEE Transactions on Knowledge and Data Engineering, 28, pp.97–106, 2014.
[4]. Tien, J. M. (Year). Big Data: Unleashing Information. Journal of Systems Science and Systems Engineering, Volume, Issue, Pages. Springer.
[5]. Chen, C. L. P., & Zhang, C. (2014). Data-Intensive Applications, Challenges, Techniques, and Technologies: A Survey on Big Data. Information Sciences, 275, pp.314–347, 2014.
[6]. Davenport, T. H. (2006). Competing on Analytics. Harvest Business Review, January 2006.
[7]. Lin, J., & Ryaboy, D. (2013). Scaling Big Data Mining Infrastructure: The Twitter Experience. ACM SIGKDD Explorations Newsletter, 14, pp.6–19, 2013.
[8]. Gartner. (2016). Analytics. Gartner IT Glossary. Available via Gartner: http://www.gartner.com/it-glossary/analytics/.
[9]. Assuncao, M. D., Calheiros, R. N., Bianchi, S. C., Netto, M. A. S., & Buyya, R. (2015). Big Data Computing and Clouds: Trends and Future Directions. Journal of Parallel and Distributed Computing, 79–80, pp.3–15, 2015.
[10]. Zachman, J. A. (1987). A Framework for Information Systems Architecture. IBM Systems Journal, 26(3).
[11]. Angelov, A., Grefen, P., & Greefhorst, D. (2012). A Framework for Analysis and Design of Software Reference Architectures. Information and Software Technology, 54, pp.417–431, (2012.
[12]. Boulon, J., et al. (2008). Chukwa: A Large-Scale Monitoring System. In Cloud Computing and its Applications, Chicago, Illinois, USA, pp.22–23, 2008.

AUTHORS PROFILE

Srinjoy Saha is a second-year student and currently pursuing Bachelor of Computer Application (BCA) from Narula Institute of Technology. He has published many research papers in reputed international journals and those are also available online. His main research work focuses on Big Data Architectures and Data Visualization. He has 2 years of research experience.

Sneha Nej is a second-year student and currently pursuing Bachelor of Computer Application (BCA) from Narula Institute of Technology. She has published many research papers in reputed international journals and those are also available online. Her main research work focuses on Big Data Search, Mining and Multidisciplinary AI. She has 2 years of research experience.

Rupa Saha is an Assistant Professor form Computer Application Department of Narula Institute of Technology. She has published many research papers in reputed international journals and those are also available online. Her main research work focuses on Artificial Intelligence & Machine Learning. She has 3 years of research experience and 17 years of teaching experience.

Debmitra Ghosh is a data scientist with research interests in machine learning, computer vision, and deep learning algorithms. She has worked in the research labs of the Indian Statistical Institute and TCS Innovation Lab. Previously she was a Business Analyst and ISTQB Certified Tester with 8+ years of experience in companies like TCS and PwC. Experienced in Performance, Automation Testing, Data Migration Testing, API testing & End-End functional testing.

Anal Rauth is a student of JIS University and currently pursuing Computer Science and Engineering. He is also a GDSC Member of JIS University. He has published some research papers in reputed international journals and those are also available online. His main research work focuses on Data Visualization and Autonomic Computing. He has 1 year of research experience.

