Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 1s (2025)

The Role of Optimization Techniques in Advancing Big Data Analytics : A Survey

E.Ramesh Babu[1], Dr.M.Sunil Kumar[2]

[1]Research Scholar, Dept.of CSE, JNTU-Anantapuramu, AP, India. rameshbabu.spmvv@gmail.com

[2]Professor, Department of CSE,School of Computing,Mohan Babu University,(erstwhile Sree Vidyanikethan Engineering College), Tirupathi, AP, India. sunilmalchi1@gmail.com

Article History:

Received: *14-08-2024*

Revised: *28-09-2024*

Accepted: *14-10-2024*

Abstract:
In the era of data proliferation, big data analytics has emerged as a vital tool for organizations seeking actionable insights from vast and diverse datasets. This paper surveys the criucial part of optimization techniques in enhancing the performance and efficiency of big data analytics. By addressing the challenges due by the volume, velocity, and variety of big data, optimization techniques enable more efficient resource utilization, faster decision-making, and improved predictive accuracy. Key applications in the areas such as healthcare, finance, smart cities demonstrate the transformative advantage of combining big data analytics with advanced optimization strategies. This survey highlights the pivotal role optimization plays in scaling data-driven insights, ensuring organizations can fully harness the power of their data.

Keywords: Big Data Analytics, Optimization

I. INTRODUCTION

Data plays a crucial role in many activities today, leading to the generation of immense volumes of data every second. The swift evolution of data across various fields underscores the need for sophisticated analytical tools that can effectively process and interpret large amounts of information.

A. Big data refers to extremely massive datasets that are complicated to manage with standard data processing tools. These datasets are defined by several key characteristics. First, volume highlights the immense amount of data. Next, velocity indicates that this data is produced at a rapid pace. Finally, variety shows that data can come in multiple formats and types, including structured, unstructured, and semi-structured forms.

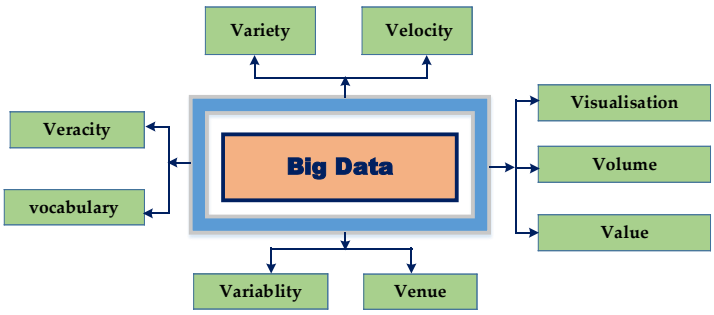

Fig1: Characteristics of Big Data

B. Issues in Big Data:

a. Privacy and security: Rapid growth of big data has brought significant advancements in various fields. However, it has also raised concerns about privacy and security. As organizations collect and store huge amounts of personal data, the risk of data breaches and misuse has increased.

b. Accessing and Sharing of Information: In today's interconnected world, information is a valuable resource that is increasingly accessible and shared. While this has led to numerous benefits, it has also raised concerns about privacy, security,

c. Analytical Challenges: As the volume, velocity, and variety of data growing drastically, organisations face significant analytical challenges. These challenges arise from the complexity of big data, the need for specialised tools and techniques, and the ethical implications of data analysis.

d. Scalability: Scalability is a crucial aspect of extensive data systems, referring to their ability to manage growing data volumes and processing requirements without appreciably sacrificing performance. As data volumes grow exponentially, scalable solutions are essential for organisations to extract value from big data.

e. Quality of Data: Data quality refers to accuracy, completeness, consistency, and reliability. High-quality data is essential for deriving accurate and meaningful insights from big data analytics. Poor data quality can lead to erroneous results, wasted resources, and compromised decision-making.

f. Heterogeneous Data: Heterogeneous data refers to diverse data in its format, structure, and source. This diversity can pose significant challenges for data analysis and integration.

g. Optimization: Optimization is finding the best solution to a problem, given constraints. It involves identifying the optimal values for variables that maximise or minimise a given objective function.

C. Motivation: The proliferation of data in our digital age has created a pressing need for innovative tools and techniques to extract meaningful insights. Big data, with its vast volume, variety, velocity, and veracity, offers a powerful solution.

D. Application Areas of Big Data Analytics:

a. **Big data in Banking**: Big data has revolutionized the banking industry, enabling financial institutions to gain valuable insights, improve customer experiences, and enhance risk management. Banks can make more informed decisions and stay competitive by leveraging the vast amounts of data generated by customers, transactions, and market trends.

b. **Finance Sector**: Big data has revolutionized the finance sector, enabling financial institutions to get more valuable insights, efficient decision-making, and enhance risk management. By leveraging the huge amounts of data produced by financial markets, transactions, and customer behavior, banks, insurance companies, and other financial firms can identify new opportunities, mitigate risks, and improve their overall performance.

c. **Telecom Sector:** Huge volumes of data are generated by the telecom sector from numerous sources. including customer interactions, network performance, and device usage. Big data analytics

has become essential for telecom providers to extract valuable insights, improve customer experience, and optimize network performance.

d. **Retail Sector:** The explosion of big data has drastically altered the retail industry. Retailers use enormous volumes of data to obtain insightful knowledge into customer behaviour, optimize operations, and improve overall business performance.

e. **Health care:** Massive volumes of data are produced by the healthcare sector, including genetic information, electronic health records, medical imaging, and patient contacts. Big data analytics offers important insights into patient care, illness prevention, and treatment efficacy that have the potential to completely transform the healthcare industry.

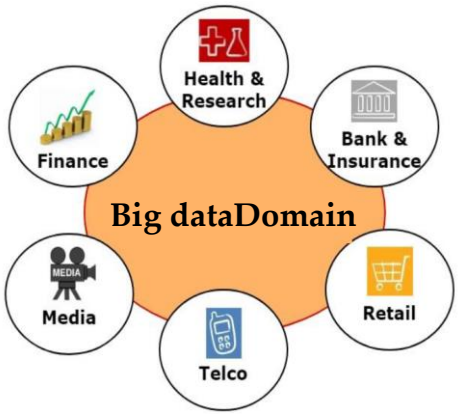

Fig2: Applications of Big Data

II. MAIN CONCEPTS

2.1 Optimization Techniques for Reliable Big Data Analytics

Optimization techniques can be applied at various stages of the big data analytics pipeline to enhance reliability and resilience. These techniques include:

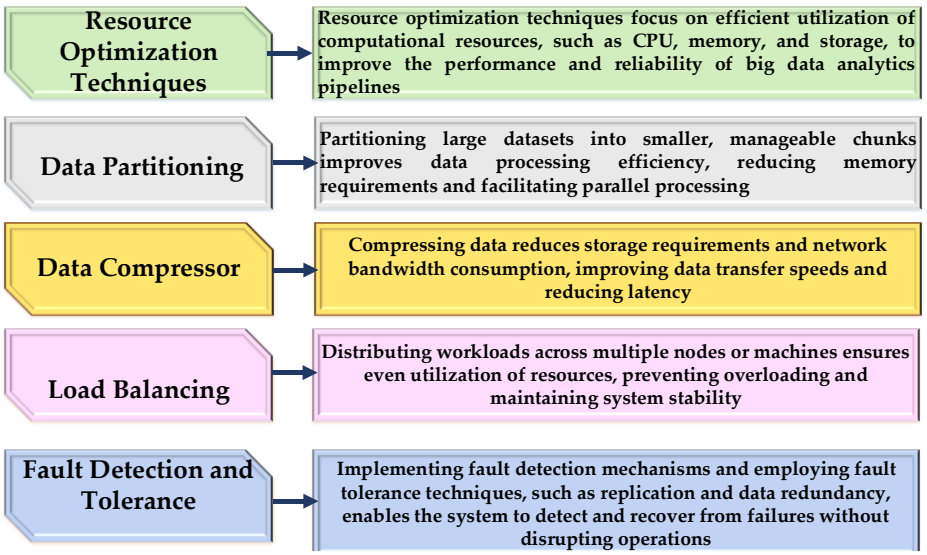

Fig3: Optimization techniques for Big data Analytics

2.2 Applications of Optimization techniques

Optimization techniques have been successfully applied in various big data analytics domains, including:

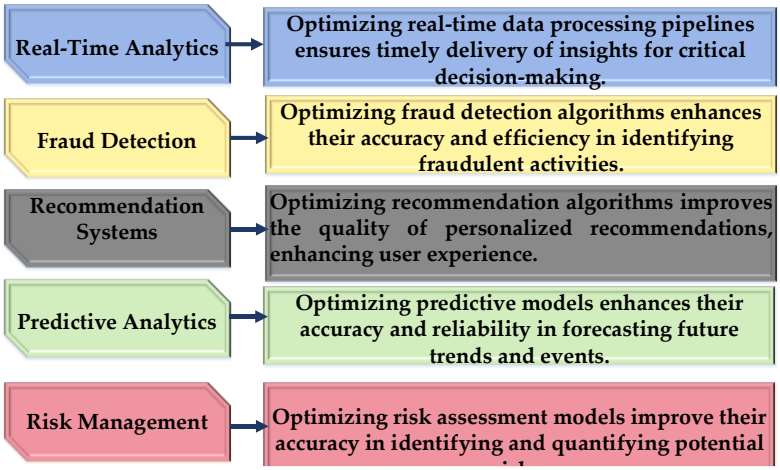

Fig4: Applications of optimization techniques for Big data Analytics

2.3 Classifications of Optimization Techniques:

A. **Resource Optimization Techniques:** These focus on the effective usage of computer resources, including memory, storage, and CPU, to raise the dependability and efficiency of big data analytics pipelines. The below are the techniques:

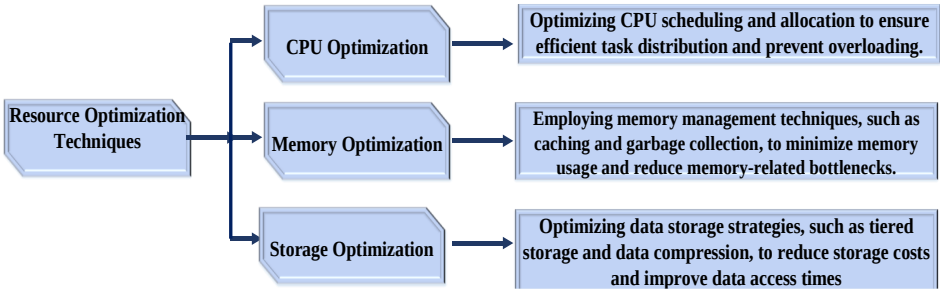

Fig5: Various Optimization Process

B. **Data Partitioning Techniques:** Data partitioning techniques divide large datasets into smaller, manageable chunks to reduce processing overhead, facilitate parallel processing, and enhance data management. These techniques include

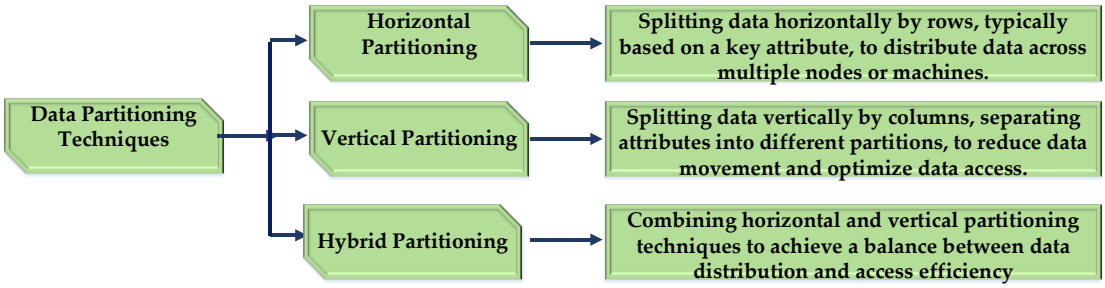

Fig6: Partitioning Techniques

C. **Data Compression Techniques:** These techniques minimise the volume of data representations to minimize storage requirements, network bandwidth consumption, and data transfer times. These techniques include:

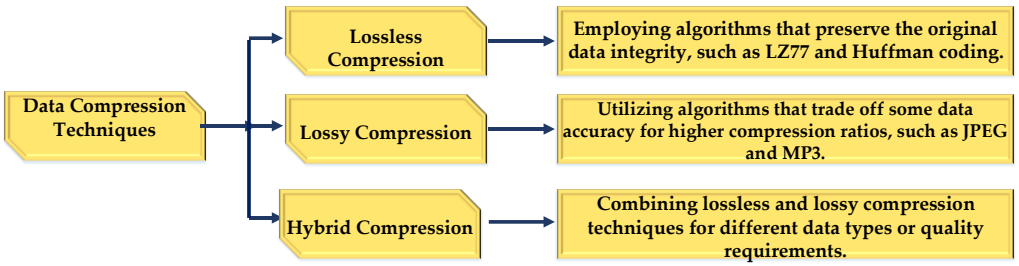

Fig7: Compression Techniques

D. **Load Balancing Techniques:** Load balancing techniques distribute workloads across multiple nodes or machines to prevent overloading, ensure even resource utilization, and maintain system stability. These techniques include:

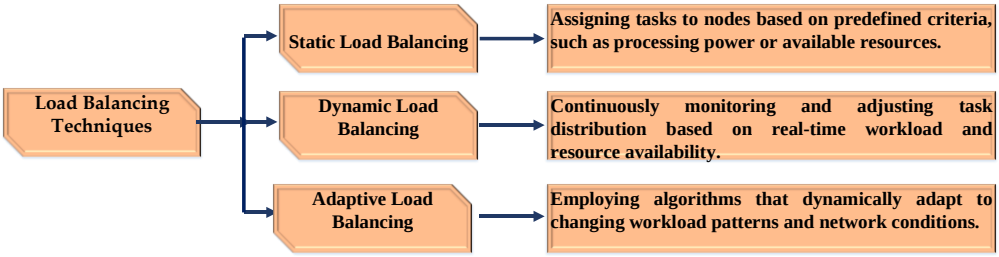

Fig8: Load Balancing Techniques

E. Fault Detection and Tolerance Techniques:

These enables the system to recognize, separate, and move past failures without disrupting operations and ensuring data integrity. These techniques include

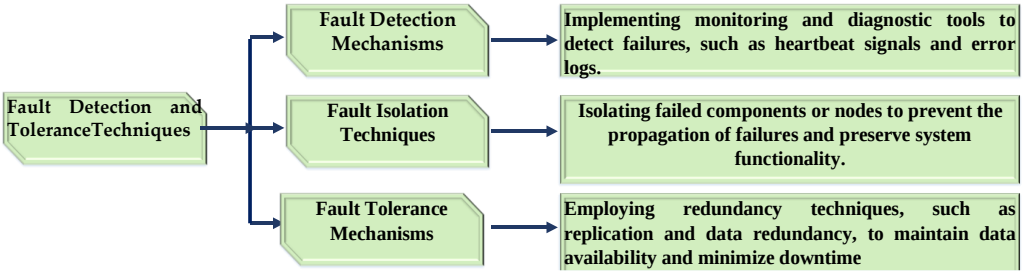

Fig9: Fault Detection and Tolerance Techniques

III. BIG DATA: CHALLENGES AND ANALYTICAL METHODS

3.1 Challenges:

While the advantages of Big Data are clear and important, several challenges must be tackled to fully unlock its potential. Some of these issues from the unique nature Big Data, while others come from current analytical methods & models. There are also limitations with today's data processing

systems. Research on Big Data challenges shows that there are difficulties in understanding how decisions are made based on the data generated and collected. Privacy issues and ethical considerations also play a role in analyzing such data. It is widely acknowledged that finding a workable solution for vast and complex data sets is an ongoing challenge for many businesses. They are continuously learning and trying out new strategies. One major concern related to Big Data is the high cost of infrastructure. The hardware needed can be expensive, even with cloud computing options available today. Additionally, human analysis often needs to be part of the process to extract important information from the data. Although computing technologies evolve quickly, the skills that business leaders require to effectively utilize Big Data are not keeping up; this presents another significant hurdle. As stated, based on the life cycle of data, the Big Data challenges can be categorized into three broad groups, they are:

- **Data challenges:** arise from features of the data itself. This includes things like volume, variety, velocity, veracity, volatility, value)
- **Process challenges:** These involve, data gathering, integrating, transforming, analysis and delivering
- **Management challenges**: address issues of ethics, governance, security, and privacy.

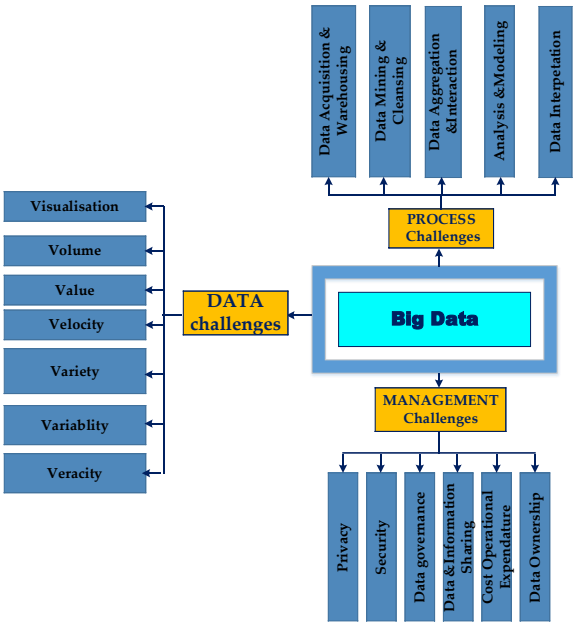

Fig9: Conceptual classification of Big Data challenges.

3.2 Big Data Analytical Methods:

Organizations must find effective methods to manage enormous volumes of diverse data. They need to convert this data into useful insights to support informed decision-making. The possibilities with Big Data are vast but are limited by the technology & tools we have for analyzing it. Big Data Analysis refers to techniques that help us study large datasets & extract intelligence from them. It can be seen as a part of a larger process of gaining insights from Big Data. To truly harness Big Data's potential and enhance business services, the right tools and methods must be effectively evaluated and categorized. The real value of Big Data emerges when we use it for decision-making. Research

shows that companies can achieve significant benefits and a competitive edge by making smart, data-driven choices. Yet, analyzing extensive datasets is not simply about tracking, classifying, or quoting data. It is more complex! Large organizations often collect Big Data and use analytics regularly to support decision-making as part of their everyday processes. Some find it challenging to include more data into their analysis efforts while still seeking to improve judgments made by the management. It can be difficult to get people, technology, and resources to align with the goals of becoming a data-driven business. But when analytical methods effectively extract insights from the data, Big Data offers opportunities to enhance decision-making and boost organizational performance.

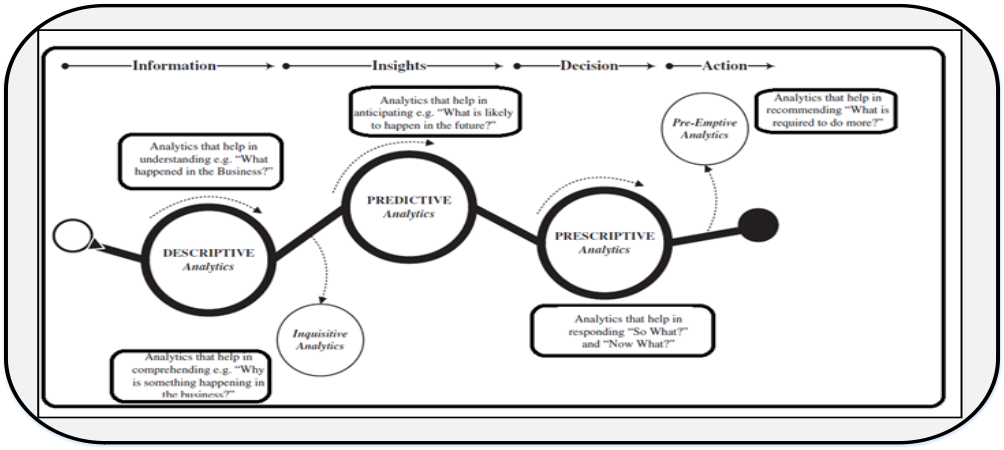

Fig10: Types of significant data analytical methods

These analytical methods help improve decision-making & boost organisational performance. They make everything clearer and easier to measure. Plus, they reveal inconsistencies, potential issues, and new opportunities. Fig. 2 shows how Big Data Analysis methods fit into five categories.

- Firstly, descriptive analytics examines data and information in detail to determine a business's existing situation. This type brings out developments, patterns, & exceptions by producing regular reports, special reports, and alerts.
- Subsequently, inquisitive analytics examines data to validate or disprove business concepts. Factor analysis, statistical analysis, and analytical drill downs are a few examples.
- Then we have predictive analytics. This one focuses on forecasting and uses statistical modeling to think about what might happen in the future.
- Prescriptive analytics centers on optimization & randomized testing. It looks at how businesses can improve service levels while cutting costs.
- Lastly, pre-emptive analytics helps organizations take early action against events that could negatively affect performance. It aims to spot potential risks & suggests ways to avoid them in advance.

IV. Optimization Implemented

Table 1 below provides a brief overview of Optimization processes along with big data analytics. It highlights characteristics of various Optimization techniques. It also describes how these methods

improve decision-making in big data domains. Several studies have explored the Optimization aspect within big data tools. Various methods for achieving Optimization are discussed here.

Table 2 shows, detailing the technique and algorithm used to meet the objectives and the specific applications where these methods were implemented.

Table 3 summarizes the previously mentioned research works based on a focus on enhancement techniques.

Table 1. Implemented Optimization Methods

Author	**Goal**	**Algorithm**	**Usecase/utilization**
Joel Wolf et.al[6]	FLEX: A Slot Apportionment Scheduling Optimizer for MapReduce Jobs	Flex allocation scheduler	Job Scheduling
Bo Dong et.al[3]	An optimised technique for storing and extracting tiny files on cloud platform.	combining many small files, prefetching for small files that are structurally related, and grouping files and prefetching for small files that are logically associated.	Tiny File distribution
Esma Yildirim et al.[8]	Pipelining, parallelism, and concurrency for application-level optimization.	Recursive Chunk Division (RCD) and Parallelism-Concurrency-Pipelining (PCP)	Map reduce optimisation
Maumita Bhattacharya et.al[9]	To improve the capacity to handle the challenges of high dimensionality and distributed data, an evolutionary algorithm is employed.	The suggested prototype reduces redundant, less promising population associates in order to display diversity through the use of educated genetic operators.	Maintaining Effective Population Diversity
Kostas Kolomvatsos et.al [10]	Appropriate Time Optimized Organization for sophisticated analysis in Big Data.	To enter incomplete outcomes, two successive decision-building prototypes are used.	Improves performance of querying big data clusters.

Table 2. Survey on Optimization techniques

Author	**Goal**	**Key Area**
Marisiddanagouda. M et al.[7]	Reduce the performance degradation caused by interaction and strong dependency between various MapReduce phases, improve Hadoop mapreduce	Optimization methods, MapReduce Performance.

	performance, and get beyond the framework's constraints.	
Dilpreet Singh et al.[32]	compares and contrasts the various hardware platforms.	Big data platforms, real-time processing, k-means clustering, graphics processing units, and scalability
Shivaraj B. G. et al. [33]	Manage several tasks within the Hadoop cluster. MapReduce schedulers manage resources by allocating MapReduce Tasks as resources.	HDFS, MapReduce, Schedulers optimisation
Sunith Bandaru et al.[1]	Many objectives are simultaneously maximized with respect to multiple variables, real-world optimization challenges.	Optimisation, Descriptive statistics, Visual data mining, Machine learning, Knowledge-driven optimisation
Hao Zhang et al.[35]	Design principles for organizing and managing in-memory data as well as useful techniques for organizing and putting into practice high-performing, effective in-memory systems.	Both practical methods for setting up and implementing high-performing, efficient in-memory systems and design principles for handling and organizing in-memory data.

Table 3. Enhancement Techniques for Optimization Process

References	**Process Capability**	**Memory Management**	**Map Reduction**	**Data Node**	**Name Node**
1	No	No	Yes	No	No
2	Yes	No	No	No	No
3	Yes	No	No	No	No
4	Yes	No	No	No	No
5	Yes	No	No	No	No
6	No	No	Yes	No	No
7	No	Yes	No	No	No
8	Yes	No	No	No	No
9	No	No	No	Yes	No
10	No	No	Yes	No	No

V. Literature Survey

When we think about data scale one big worry is processing. It's to have good performance even when handling huge amounts of data. We focus on something called scale independence, which aims to improve results when dealing with large datasets. This means we can figure out how much data is necessary to answer questions without worrying about the total size or changes in the dataset. There are many ways to achieve scale independence in big data. These approaches work regardless of size

or range. Typically, most database methods choose views that can speed up how queries run on average.

References	Objective
[11]	This article presents a method for selecting and maintaining views that are scale-independent. It uses new static analysis strategies. These strategies help ensure that any new views created do not lead to inefficiencies as they grow. By applying unique static algorithms, we can avoid scaling issues. These algorithms ensure that views maintain certain properties relating to their scale and the effort needed for ongoing maintenance. That said, there is a challenge: as the costs for updating Incrementally Maintained Views (IMV) rise with application size, it affects how we gather these views.
[12]	Relationship cardinality and associated size specifications are described using the simplest SQL extension. To avoid output deterioration, the dba can select a normal constrained scheme over an infinite scheme. In other words, the compiler makes every effort to bind the computation if it is difficult to provide a constrained scheme for a given task. Strict limitations are imposed by PIQL on the quantity of I/O operations required to respond to any query.
[13]	You can describe connection cardinality and the associated size specifications with the most basic SQL extension. To avoid output deterioration on such requests, the dba selects a normal constrained scheme over an infinite scheme. In other words, the compiler makes every effort to bound the computation if it cannot produce normal constrained scheme for a given task.The amount of I/O operations required to answer any inquiry is limited by PIQL in [13].
[14]	Schema's accessible sections are established for processing, and a detailed analysis is conducted when responding to a query that has integrity and access restrictions. Super-extractability is a query-independent quality that is highlighted in the actual schema. If we can use the intended access pattern to extract a superset of values from a schema, then the schema is considered super-extractable. A query is considered logically accessible rewritable under constraints and circumstances if it can be verified by executing it on the accessible data.
[15,16]	The approaches investigate methods for responding to queries with restricted access patterns. In particular, the section that addresses rewriting queries with integrity constraints and restricted access patterns. A technique is given to find a precise executable plan (if any). The binding pattern of a relation is used to calculate the full response to a query. The maximum number of characteristics that can be utilized to access a relation is defined by its binding pattern.
	Proposed a Scalable Consistency Adjustable Data Storage (SCADS) architecture to allow users to specify requirements specific to their applications. Furthermore, machine-learning techniques are utilized to address problems with performance and find the resource needs of future queries before to their execution. Query Language with Performance Safety, –Three methods that incorporate data scale independence are Performance Tradeoff, Declarative Consistency, and Scaling Up and Down Using Machine Learning. Effective web programming is made possible byPQL, which guarantees predictability and scalability. Developers can

[17,18]	assess an application's accuracy with respect to required performance SLAs (Service Level Agreements). The capacity to effectively achieve SLAs by adding and subtracting capacity using machine-learning models is known as "upgrading and downgrading" machine learning.
[19]	HadoopDb consists of two layers: data processing and storage. The block structure framework in the storage layer controls the core name node. Using a given partition key, the data loader component divides the data worldwide. The task tracker and the separate database structure on the nodes are connected via a database connector provided by HadoopDb.
[20]	Hadoop is an enhanced version of the map-reduce architecture that makes use of cache technologies to make the job scheduler loop-aware, extending map reduction by providing support for iterative queries in programming languages. It provides an exclusive distributed and parallel architecture for massively parallel iterative data processing applications.
[21]	The method suggests an optimization framework for map-reduce tasks that resemble SQL. A descriptive query language that can be better optimized and captures most of the processing in declarative form. It explains how the algebraic forms that are taken out of queries are mapped to the optimizer..
[22]	Numerous algebraic optimizations are also covered, including summarizing the combined function structure from the reduced function of a map-reduced task and fusing map-reduced task into one task. In deductive,relational database designs, an incremental estimation approach is given to compute the view obtained from the relations. For every derived tuple in the view, the number of alternative derivations is tracked using a counting technique. The method is applicable to both set and duplicate semantics during query evaluation. This paper explores a different method of negation and aggregation for non-recursive views. Because the method computes the result precisely, it produces an optimal outcome.
[23,24,25]	Recursive delta-based computation is discussed more in-depth in the work [23]. Iterative computations between iterations are mostly made easier by the delta adjustments, and the state can be readily and extensibly updated. In the runtime environment of REX, these queries are implemented and optimized using a programming paradigm that is presented in this paper. Failures are handled graciously in the REX runtime mechanism. In order to remove redundancies, Comet [24], a cost-based optimizer, talks about those shared calculations at the SQL and Map Reduce levels. A rule-based optimizer was defined [25].
[26,27]	[26] describes the architecture of useful map-reduction techniques for related joins, deep learning, and data processing. The operations of the Map Reduce are frequently carried out on a stand alone node that has preprocessed data prior to calling the map functions. A detailed analysis of HiveQL, a declarative language that resembles SQL, may be found in [27].
[28]	Described a megastore that has the statistics and schema required for query compilation, data exploration, and optimization. The execution plan is created by the query generator utilizing the Meta store's information. At last, all of the jobs are finished in the dependent sequence. The task is done only once the prerequisites for any dependent tasks have been fulfilled. The part of the plan that

	is serialized into an XML file called plan.xml is done via a map/reduce job. Hadoop instances of Exec Reducers and Exec Mapper were generated, and it was put to the job tracker cache. The gradual maintenance view is discussed in [28].
[29]	The query calculus is inherited with the properties of a ring that computes inverses for delta queries and has a regular form for polynomials. It also eliminates costly query operators, such as joins, which are required to compute incremental view maintenance. The technique in [29] offers an illustration of a novel approach and goes into considerable detail regarding managing large-data problems and data-intensive processing.
[30]	It is better to avoid haphazard data access and to sequence data processing computations. Data center dependability is covered in this essay. A scalable distributed architecture for learning models from large datasets is included in the recommended technique [30].
[31]	A novel architecture known as Spark was put forth in [31] and offers applications for preserving the fault tolerance and scalability of MapReduce. It's a collection of read-only items divided over several computers that can be recovered in the event that one is destroyed. An acyclic data flow model is employed by most technologies to run large-scale data-intensive applications, yet this paradigm is inefficient for these kinds of applications. The main topic of this work is data reuse via several concurrent procedures.
[34]	Suggested a machine learning-based edge-cloud computing big data analytics architecture that is tailored for the IoT. The suggested plan integrates cloud technology with an edge intelligence module to effectively handle and store massive data at the network's edges. The suggested plan consists of two layers: cloud processing and IoT edge. For effective cluster management, an optimized YARN is employed. Apache Spark is used to experimentally mimic the suggested data design using a real data set. The conventional processes and current proposals adorn the comparative analysis. The outcomes support the effectiveness of the work we suggested.
[36]	The survey had given extensive review on optimization strategies for dependable and robust big data analytics. The survey addresses why big data analytics should be optimized for robustness and dependability, provides a thorough categorization of optimization methods, assesses each method's efficacy, and highlights new developments and interesting research avenues in the area. The primary conclusions drawn from the survey underscore the significance of fault detection and tolerance mechanisms, load balancing, data compression, resource optimization, and data partitioning in augmenting the robustness and dependability of big data analytics pipelines. The report also emphasizes the need for self-optimizing and adaptive strategies that can adapt to changing situations dynamically and maximize the use of available resources.
[37]	The GTOA-MLBDA technique for big data categorization is developed in this publication. Big data is categorized using the GTOAMLBDA technique, which the Map Reduce tool can handle. Using GTOA, the GTOA-MLBDA technique lowers dimensionality, identifies key features, and improves classifier performance. Furthermore, the FLNN model is used for big data classification.

	Lastly, the Adam optimizer is used to fine-tune the FLNN model's parameters. Big data datasets were subjected to a thorough experimental examination, and the findings demonstrate that the GTOA-MLBDA technique's combination of the GTOA and FLNN classifier produced better outcomes than traditional methods.
[38]	Big data, the ant colony optimization technique, and the query process are all covered in this work. The ant colony optimization technique is used to big data, which manipulates its data using hardware, software, and tools. Understanding hybrid ACO Algorithm approaches for distributed database query optimization is still a very new area of study. Many studies on the design and implementation of ACO hybrids to address various optimization problems are now being conducted. The results indicate that ACO hybrids can be useful and realistic in optimization-related tasks. Furthermore, the investigators disclosed that the execution showcased pragmatic approaches about the use of algorithms, along with the frequency with which the database administration system broadened and approached the query to match its magnitude. Even with all of the experiments, there is still a great deal of potential to improve search algorithms and get the best answers when utilizing ACO hybrids for distributed database queries, particularly when numerous constraints are changed and the associations' size and complexity grow.
[39]	The work has combined ML and DL in big data analytics applications and systems. IoT sensor devices are producing massive amounts of data that are beyond human capacity to interpret, such as structured (like tables) and unstructured (like text and images). Nevertheless, research on massive data analysis for IoT without jeopardizing IoT privacy is lacking. A distributed learning strategies were discussed that work well with IoT infrastructures. Next, in order to identify trends and gain knowledge from IoT data, presented unique dynamic deep learning method and a privacy-preserving distributed learning framework. To demonstrate the efficacy and efficiency of our system, simulations were done. Planned to use new deep learning tehniques, apply differential privacy to the IoT learning framework, and build the learning framework in TensorFlow2 or PyTorch in the future.

VI. ANALYSIS

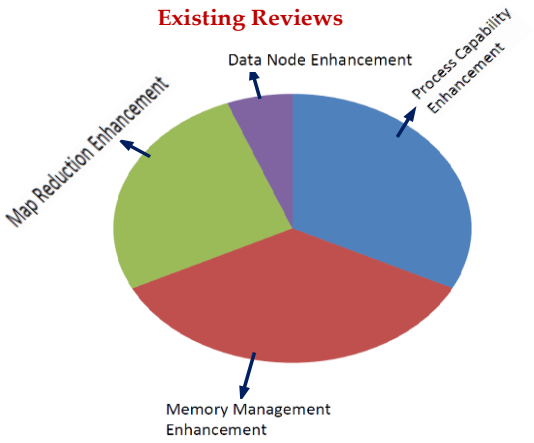

Fig11: Review on Different Enhancement techniques

6.1 **Process Capability Enhancement**

6.1.1 Level Optimization: Performance has been enhanced by process capability development, according to study on Big Data Application-Level Optimization Transfers across Pipelining, Parallelism, and Concurrency[8]. Scientific cloud applications require application-level transfer adjustment parameters like pipelining, parallelism, and concurrency to get around data transmission constraints. Optimizing these factors can lead to the attainment of an optimal transmission rate. According to the research, optimization algorithms can provide consistent growth to maximize throughput on intra- and inter-cloud transfers by using these representations and principles..

6.1.2 Evolutionary optimisation*:* The study that follows focuses on improving process capabilities. The suggested model makes use of knowledgeable genetic operators to add diversity. Additionally, a technique for handling high dimensionality is presented in the research. The primary goal of the algorithm presented in this work is to handle advanced-dimensional problem domains. High-dimensional optimization problems, especially those with intricate multi-model solution spaces, can be handled by the POPULATION_EA prototype[2]

6.1.3 Optimized Job Allocation Scheme*:* Flex is a clever and adaptable MapReduce load distribution mechanism. It is adaptable in that it can improve any of the common scheduling indicators, including sales, average response time, stretch, and deadline-based punishment functions. It can achieve performance that is close to the theoretically optimal, which makes it intellectual.

6.2 Memory Management Enhancement

6.2.1 In-Memory Big Data Optimization*: :* Memory Management Enhancement is demonstrated in this survey on In-Memory Big Data Management and Processing. The study provides a thorough examination of key technologies for memory management [32] and a review of related literature. The design values for in-memory data handling and management, as well as useful techniques for organizing and carrying out efficient and functional systems, are the focus of this survey.

6.3. Map Reduction Enhancement

6.3.1 Platform optimization according to Big Data*:* For a given application, the choice of platform is usually influenced by factors like model development, speed or throughput optimization, and data quantity. Various big data platforms [32] that offer different features; finding the exact platform necessitates thorough knowledge of each platform's capabilities.

6.4 Data Node Enhancement

6.4.1 Knowledge discovery in multi-objective optimization: Data node augmentation is discussed and approaches for knowledge discovery in multi-objective optimization. A population of randomly generated outcomes or entities is the starting point for maximum multi-objective optimizers [1].

6.5 Name Node Enhancement

6.5.1 Small files optimization*:* In their study on storing and retrieving tiny files, Bo Dong et al. developed an improved technique to boost small file storage and retrieval capabilities on HDFS. In this study, it has been suggested to merge multiple tiny files, prefetch for structurally connected small files, group files, and prefetch for logically linked small files.

VII. CONCLUSION

Big data analytics is now a vital tool for organisations looking to extract valuable information from enormous, intricate datasets. This survey examined the core ideas, instruments, and processes that underpin big data analytics along with optimization strategies maximize its efficacy. The ability to process, analyze, and interpret enormous volumes of data in real-time has been greatly enhanced by the incorporation of optimization methods, which range from conventional techniques like linear programming to sophisticated machine learning algorithms.More effective data processing, storage, and analysis are made possible by contemporary optimization approaches, which handle important issues including data volume, velocity, and diversity. More precise decision-making, resource allocation, and predictive modeling in different industries, like healthcare, finance, and smart cities, are made possible by the synergy of big data analytics and optimization.

Future developments in fields like edge computing, quantum computing, and AI-driven optimization will probably improve big data analytics even further. As businesses depend more and more on data-driven strategies, optimal big data analytics will play a critical part in expanding the possibilities of what can be done with data at scale.

References

[1] Bandaru, S., Ng, A. H., and Deb, K. Data mining methods for knowledge discovery in multi-objective optimization: Part a-survey. Expert Systems with Applications 70 (2017), 139-159.

[2] Bhattacharya, M., Islam, R., and Abawajy, J. Evolutionary optimization: a big data perspective. Journal of network and computer applications 59 (2016), 416-426.

[3] Dong, B., Zheng, Q., Tian, F., Chao, K.-M., Ma, R., and Anane, R. An optimized approach for storing and accessing small files on cloud storage. Journal of Network and Computer Applications 35, 6 (2012), 1847-1862.

[4] Gu, R., Yang, X., Yan, J., Sun, Y., Wang, B., Yuan, C., and Huang, Y. Shadoop: Improving mapreduce performance by optimizing job execution mechanism in hadoop clusters. Journal of parallel and distributed computing 74, 3 (2014), 2166-2179.

[5] Hua, X., Wu, H., Li, Z., and Ren, S. Enhancing throughput of the hadoop distributed file system for interaction-intensive tasks. Journal of Parallel and Distributed Computing 74, 8 (2014), 2770-2779.

[6] Wolf, Joel, Deepak Rajan, Kirsten Hildrum, Rohit Khandekar, Vibhore Kumar, Sujay Parekh, Kun-Lung Wu, and Andrey Balmin. "Flex: A slot allocation scheduling optimizer for mapreduce workloads." In Middleware 2010: ACM/IFIP/USENIX 11th International Middleware Conference, Bangalore, India, November 29-December 3, 2010. Proceedings 11, pp. 1-20. Springer Berlin Heidelberg, 2010.

[7] Mr. Marisiddanagouda. M, M. R. M. Survey on performance of hadoop map-reduce optimization methods. International Journal of Recent Research in Mathematics Computer Science and Information Technology 2 (2015), 114-121.

[8] Yildirim, Esma & Arslan, Engin & Kim, Jangyoung & Kosar, Tevfik. (2015). Application-Level Optimization of Big Data Transfers Through Pipelining, Parallelism and Concurrency. IEEE Transactions on Cloud Computing. 4. 1-1. 10.1109/TCC.2015.2415804.

[9] Maumita Bhattacharya, Rafiqul Islam, Jemal Abawajy,Evolutionary optimization: A big data perspective,Journal of Network and Computer Applications,Volume 59,2016,Pages 416-426,ISSN 1084-8045,https://doi.org/10.1016/j.jnca.2014.07.032.

[10] Kolomvatsos, K., Anagnostopoulos, C., and Hadjiefthymiades, S. An efficient time optimized scheme for progressive analytics in big data. Big Data Research 2, 4 (2015), 155-165.

[11] Armbust M, Liang E, Kraska T, Fox A, Micahel J, Franklin DA Patteson," Generalized scale independence through incremental precomputation", Proceedings of Special Interest Group on Management Of Data, 2013, pp. 625–636.

[12] Armbrust M, Curtis K, Kraska T, Fox A, Franklin MJ, Patterson DA, "PIQL: Success-tolerant query processing in the cloud", Proceedings of the Very Large DataBase Endowment, Vol. 5, Issue 3, 2011, pp. 181–192.
[13] Armbrust M, Tu S, Fox A, Franklin MJ, Patterson DA, "PIQL: A Performance Insightful Query Language", Proceedings of the International Conference on Management of Data, 2010, pp.1207– 1210.
[14] Deutsch A, Ludascher B, Nash A. Rewriting queries using views with access with access patternsunder integrity constraints. Lecturer notes in Computer Science. 2005;3363:352–67.
[15] Chen.Li. Computing complete answers to queries in the presence of limited access patterns ". The Very Large DataBase Journal. 2003;12(3):211–27.
[16] Fox,D.A.Patterson.N.LanhamMArmbrust,A,B.Trusshkowsky,J.Trutna and Haruki.Oh,"SCADS:Scaleindependent storage for social compuitng applications",Proceedings of 4rth Biennial conference oninnovative data systems research,2009.
[17] Abouzeid A, Bajda-Pawlikowski K, Abadi D, Silberschatz A, Rasin A, "HadoopDB: an architecturalhybrid of MapReduce and DBMS technologies for analytical workloads", Proceedings of Very LargeDataBase Endowment,2(1):922–933, Aug. 2009.
[18] Bu Y, Howe B, Balazinska M, and M. D. Ernst,"HaLoop: efficientiterative data processing on large clusters", Proceedings of Very Large DataBase Endowment 3(1–2)pp.285–296, Sept. 2010.
[19] Ekanayake J, Li H, Zhang B, Gunarathne T, Bae S-H, Qiu J, andG. Fox, "Twister: a runtime for iterativeMapReduce, Proceedings of High Performance and Distributed Computing",pp. 810–818, 2010.
[20] Fegaras L, Li C, and U. Gupta,"An optimization framework for Map-Reduce queries",Proceedings ofExtending Database Technology, pp. 26–37, 2012.
[21] Gupta A, Mumick IS, and V. S. Subrahmanian,"Maintaining views incrementally", Proceedings ofSpecial Interest Group on Management Of Data, pages 157–166, 1993.
[22] Mihaylov SR, Ives ZG. and S. Guha,"REX: Recursive,delta-based data-centric computation",Publication of Very Large DataBase,5(11):pp.1280–1291,2012.
[23] He B, Yang M, Guo Z, Chen R, Su B, Lin W, and L. Zhou,"Comet: batched stream processing for data intensive distributed computing",Proceedings of Symposium On Cloud Computing, pp.63–74, 2010.
[24] Lee R, Luo T, Huai Y, Wang F, He Y, and X. Zhang,"YSmart: Yetanother SQL-to-MapReduce translator", Proceedings of International Conference on Distributed Computing System, pp.25–36, 2011. Page 9/9
[25] K. Shim,"MapReduce algorithms for big data analysis", Publication of Very Large DataBase,5(12):pp.2016–2017, 2012.
[26] Thusoo A, Sarma JS, Jain N, Shao Z, Chakka P, Zhang N, Anthony S, Liu H, and R. Murthy. Hive - petabyte scale datawarehouse using Hadoop|, Proceedings of. International Conference on Data Engineering, pp.996–1005, 2010.
[27] Christoph, Koch,"Incremental Query Evaluation in a Ring of Databases",Proceedings of Principle Of Database Systems,pp.87–88,2010.
[28] Lin J and C. Dyer,"Data intensive text processing with MapReduce",Synthesis Lectures on Human Language Technologies,pp.177,2010.
[29] Panda B, Herbach JS, Basu S. and R. J. Bayardo,"PLANET:massively parallel learning of treeensembles with MapReduce". Publication of Very Large DataBase. 2009;2(2):1426–37.
[30] Zaharia M, Chowdhury M, Franklin MJ, Shenker S, andI. Stoica,"Spark: cluster computing with working sets",Proceedings of HotCloud, pp.10–10, 2010.
[31] Roy, Chandrima, Siddharth Swarup Rautaray, and Manjusha Pandey. "Big Data Optimization Techniques: A Survey." International Journal of Information Engineering & Electronic Business 10, no. 4 (2018).
[32] Singh, D. and Reddy, C.K., 2015. A survey on platforms for big data analytics. *Journal of big data*, *2*, pp.1-20.
[33] Shivaraj B. G., N. N. Survey on schedulers optimization to handle multiple jobs in hadoop cluster. International Journal of Science and Research 4 (2013), 1179-1184.
[34] M. Babar, M. A. Jan, X. He, M. U. Tariq, S. Mastorakis and R. Alturki, "An Optimized IoT-Enabled Big Data Analytics Architecture for Edge–Cloud Computing," in *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 3995-4005, 1 March1, 2023, doi: 10.1109/JIOT.2022.3157552.
[35] Zhang, Hao & Chen, Gang & Ooi, Beng & Tan, Kian-Lee & Zhang, Meihui. (2015). In-Memory Big Data Management and Processing: A Survey. IEEE Transactions on Knowledge and Data Engineering. 27. 1-1. 10.1109/TKDE.2015.2427795.

[36] El Baqqaly et al, Mesopotamian,"Optimizing Big Data Analytics for Reliability and Resilience: A Survey of Techniques and Applications", Journal of Big Data Vol. (2023), 2023, 118–124.
[37] Prabaharan, G. & Sudharsanam, Sesha & Venkatesh, K. & Sundararajan, Suma & G, Shobana. (2023). Group Teaching Optimization Algorithm with Machine Learning Model for Big Data Analytics. 147-152. 10.1109/ICSCNA58489.2023.10370497.
[38] Deepak Kumar, Dr. Vijay Kumar Jha," An Enhanced Query Optimization Technique In Big Data Using Aco Algorithm", International Research Journal of Engineering and Technology (IRJET) Volume: 08 Issue: 11 | Nov 2021, e-ISSN: 2395-0056 p-ISSN: 2395-0072.
[39] M. Guo, N. Pissinou and S. S. Iyengar, "Privacy-Preserving Deep Learning for Enabling Big Edge Data Analytics in Internet of Things," 2019 Tenth International Green and Sustainable Computing Conference (IGSC), Alexandria, VA, USA, 2019, pp. 1-6, doi:10.1109/IGSC48788.2019.8957195.