International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V71I6P233

Volume 71 Issue 6, 335-345, June 2023
© 2023 Seventh Sense Research Group®

Original Article

Advanced Machine Learning Based File Storage Model for Hadoop Dynamic File Access in Bigdata Analytics

Yallapragada Ravi Raju[1], D. Haritha[2], R. Phani Vidyadhar[3], Mohammed Ayad Alkhafaji[4]
K saikumar[5], Ahmed J. Obaid[6]

[1,2]Department of Computer Science and Engineering (CSE), Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur District, Andhra Pradesh, India.
[3]Vardhaman College of Engineering, Shamshabad Rd, Kacharam, Telangana
[4]National University of Science and Technology, Dhi Qar, Iraq
[5]School of Engineering, Department of CSE, Malla Reddy University, Maisammaguda, Dulapally, Hyderabad, Telangana
[6]Faculty of Computer Science and Mathematics, University of Kufa, Iraq

[1]Correspond Author : ravi.y40@gmail.com

Received: 06 February 2023 Revised: 31 May 2023 Accepted: 18 June 2023 Published: 25 June 2023

Abstract *- Modern technologies manage bigdata data storage applications, which improves application storage. Bigdata surveys simplify file storage. The survey finds no viable file management mechanisms. Existing approaches store unstructured and organized files insecurely. Bigdata analytics requires complex file management. This study implements map secure reduce layers (MSR) and elastic net regression (ENR). MSR-ENR approaches are tested using the HDFS file-handling infrastructure. MSR-ENR can handle all memory file types and extensions (. dox, docs, .pdf, .rar, etc..). Finalize processing time, sensitivity, accuracy, throughput, and recall. This MSR-ENR approach surpasses simulations, challenging existing technology. Big data platforms maintain the cloud, servers, and Hadoop. Data-driven Hadoop modelling cannot provide dynamic actions. App weaknesses include latency and storage. Big data platforms and the Internet have not guided cloud storage upkeep. Big data cloud gateways will drive development and change. This study displays the current method (DL-enabled operational Facilities) through sponsorship software. Intelligent closed-loop video surveillance may speed up and improve large data file maintenance. This speeds up cloud-based large-file production. U-net uses Hadoop and Sparks to analyze data. This software uses Python 3.7. This U-net big data analytics software is competitive.*

Keywords *- Big data, Hadoop, MSR, ENR, DL-enabled operational Facilities.*

1. Introduction

Internet-based social network platforms are progressively permeating people's everyday lives, places of work, and places of education and have emerged as the primary location and significant source of social information dissemination. Unstructured social data increases as online social networking platforms continue to expand [1]. Every user is a data source on social networks, & the amount of information is expanding rapidly, and its worth is obvious. If we can grasp more knowledge in today's information-driven world, we can take advantage of the market. Despite this, the platform can still not understand its acquired raw data. Using data visualization technologies, it is possible to represent a wide range of data and convey information in a short period of time. Businesses are interested in how visualization technology can be used to quickly and effectively discover and convert significant data from vast, complex social network datasets into information that can be readily comprehended by users and systems alike [2].

Data visualization is now being investigated by researchers in related domains, and some theoretical findings have already been obtained. Virtual reality techniques may be used to create 3D data visualizations, according to Kline and Volegov. Using VRT, information, types of equipment, and experimental settings may be shown in three dimensions instead of only two dimensions on the flat screen. Using 2 VR software tools, the unity gaming engine and A-Frame, the visual interface of data and high-energy physical goals may be achieved. Virtual reality technology may be used to produce a 3D data visualization in this way. Mrsic et al. recommended social network analysis and data visualization technologies for knowledge dispersion studies. For social media network research, Fb is used to get information on public page relationships. Social media group communication is carried out via the creation of fundamental models, the acquisition and processing of information, and finally, the display of results. This strategy does have some validity. These previous techniques have not addressed all of the problems with unstructured data on social networks [3].

Using a visual dynamic simulation environment of unorganized social media text, these problems may be addressed. Data from social networks is extracted using an adaptive threshold that changes depending on how it has been collected and prepared. In order to exhibit unstructured data in social networking sites, a visible, dynamic analysis of unorganized data is built using Hadoop clustering architectural and data visualization technologies, and HDFS is used for data durability. MapReduce may be used for global computations in visual data feature classes [4]. This technique displays unstructured social network data well, consistently, and efficiently.

1.1. Hadoop Platform

A web crawler called Nutch is the basis of Hadoop's distributed system architecture. Unstructured data storage and computational solutions in Hadoop are especially well-suited for offline analysis of massive amounts of data. [5]. Data storage and computing are handled by the core of the open-source model, which runs on a massive computer cluster.

1.1.1. HDFS

Hadoop Distributed File System is another name for HDFS. As a result, the GFS was born, defined by scattered dealing out of information, high dependability, and high availability. Files larger than a few terabytes may be stored in HDFS, allowing streaming data access and the "write once, read many times" read-write mode. HDFS also allows for hardware faults and supports extremely big files (up to several terabytes). Machine node requirements for Hadoop clusters are quite low. In the event of a hardware failure, a redundancies backups using HDFS is available that is totally undetectable to the user. Figure 1 illustrates the master-slave structure used by Hadoop [6].

HDFS includes One Name Node (ONN), One Secondary Name Node (OSNN), and a Number of Data Nodes (NDN). HDFS has a default file size of 64MiB. The client must communicate with the Name-Node to read and write data. The Name-Node stores a Name-Space to retain the cluster's data storage. As part of its data redundancy strategy, HDFS stores identical data blocks across several servers and racks [7].

1.1.2. MapReduce

It is a structure for pressing unstructured data in parallel using the MapReduce programming language. A huge data set is divided among the cluster's nodes, where each node does a portion of the work, and the intermediate results are combined to get the final results [16–18]. When it comes to solving complicated issues in parallel, MapReduce is an excellent choice. Figure 2 depicts the MapReduce mechanism in action.

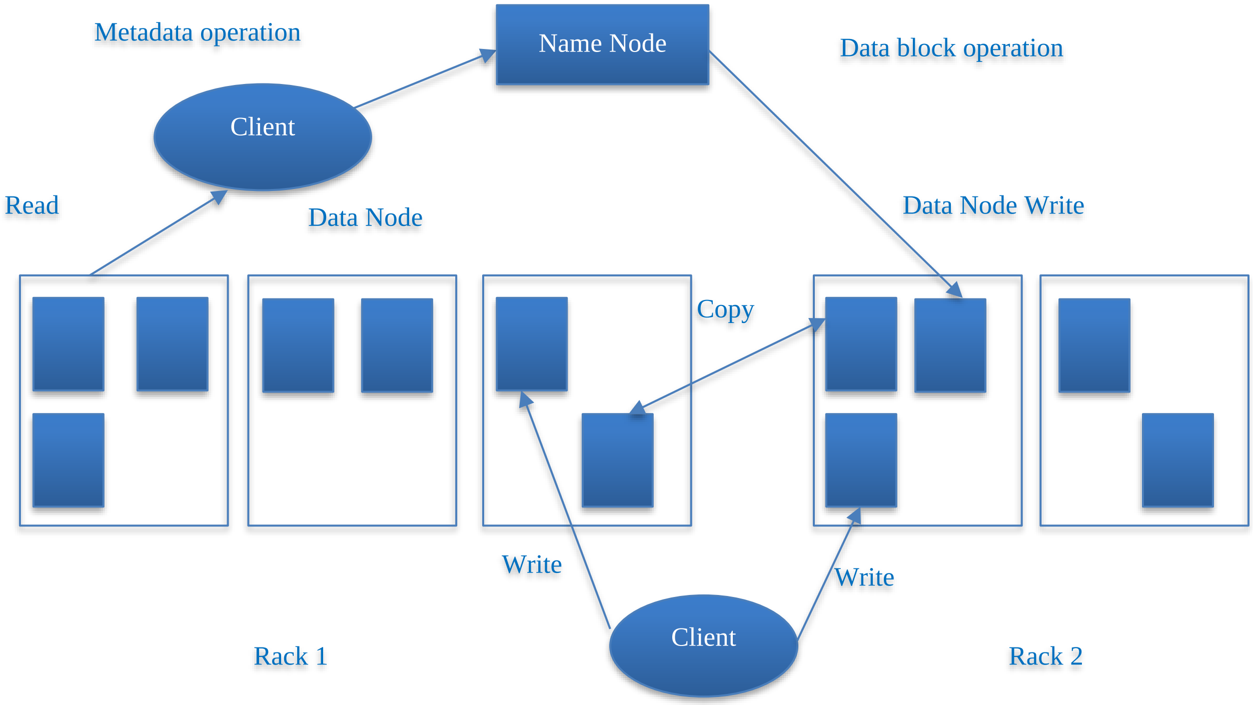


Fig. 1 Architecture of HDFS

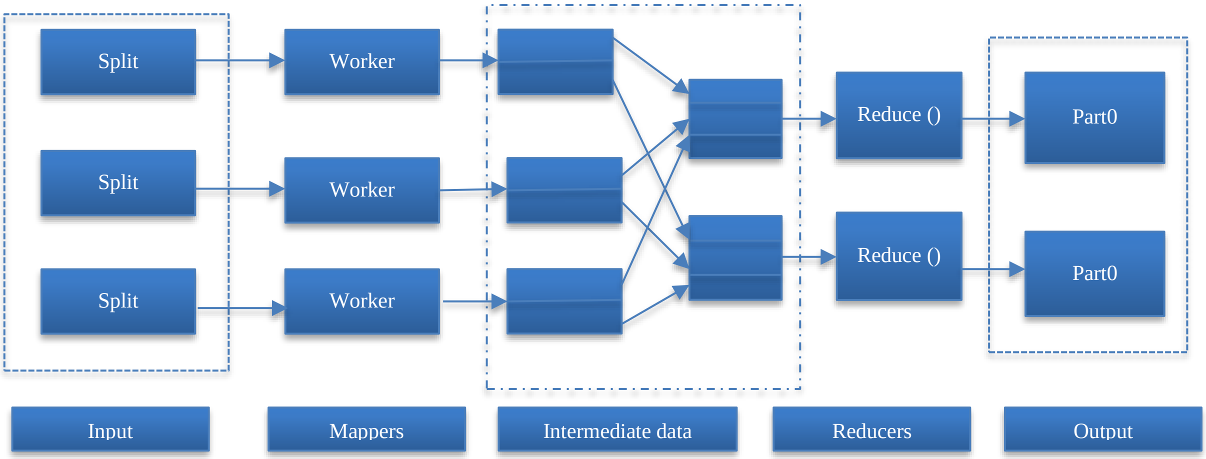


Fig. 2 MapReduce flowchart

The map task sends data to the reduce task using a pull model. The reduction with HTTP requests to obtain important data from each map job, which is stored locally as an interim sample calculation.

2. Literature Survey

Chen, H. M., et al. [2016] Massive dynamic BIMs may now be stored in Bigtables using a new format. An Apache Hadoop component called MapReduce managed huge amounts of information from dynamic BIMs. For diverse applications, researchers in this study presented a MapReduce distributed computing architecture for big data analysis to efficiently collect and compute necessary information from dynamic BIMs stored in the data centre. This paper outlines the theoretical foundations of the proposed framework, as well as the results of an experimental evaluation. The findings showed that the suggested framework could manage enormous BIMs in a scalable and reliable manner.[1]

Ahad, M. A., et al. [2018] All of this is done before the files are sent onto the Hadoop distributed file system so it is safe and secure. The little files are combined into a single unit for more efficient storage. Here, "dynamic merging procedures" rather than a generic merging strategy are the primary criterion for merging tiny-size files. As an added bonus, the idea uses the Software Defined Networking (SDN) concept to route files from their origin to their destination more efficiently. For the suggested architecture to save Namenode memory overhead and reduce disc seek time, the empirical findings demonstrate that it is beneficial to use the proposed architecture[2]

He, Y. et al. [2011] When it comes to traditional Web service providers and social networking sites, MapReduce-based data storage processes are crucial for quickly and accurately analyzing user patterns and needs using big data analytics (e.g., Facebook). Several factors may affect a system's ability to function, including data placement in its warehouse. High data transmission data loading and query processors, high storage space economies, and excellent adaptability to highly dynamic workloads were some of Facebook's production system criteria for our optimal scheduling structure. We studied the three most prevalent storage management formats using MapReduce to analyze enormous volumes of data: rows, columns, and hybrids. We demonstrate that they are not well-suited to distributed systems' handling of large amounts of data. An implementation of RCFile (Record Columnar File) in Hadoop is shown here. In a series of exhaustive tests, we demonstrate RCFile's ability to meet all four criteria. Facebook's data warehouse system uses RCFile as the default choice. Hive and Pig, two of Facebook's and Yahoo's most frequently used data analysis platforms, have both incorporated it.[3]

Ghazi, M. R et al. [2015] Data analysis, examination, and processing of an enormous quantity of unstructured information have been a difficult task. MapReduce and the Hadoop Distributed File System are the two main components of Hadoop, and they are discussed in depth in this article (HDFS). JobTrackers and TaskTrackers handle the monitoring and execution of jobs in the MapReduce engine. NameNode, DataNode, and Secondary NameNode are all components of HDFS, a distributed file system designed to handle distributed storage efficiently. It is possible to utilize the information supplied to create large-scale distributed applications that use several nodes' processing capabilities.[4]

Krish, K. R et al. [2016] The emergence of heterogeneous storage systems (such as RAMDisks) alongside regular HDDs is a fascinating new trend in storage management for massive data structures like Hadoop or Spark. The more processing and storage resources an application uses, the more difficult it gets to plan data accesses and requests to a suitable storage device. To improve overall application I/O performance, Dynamic Data Management for Data Processing Frameworks (DUX) is an application-tuned solution that uses SSDs only for those applications that would benefit the most. [5]

Herodotou, H. et al. [2011] Starfish, a new big data analytics platform with self-tuning capabilities, has just been launched. With the help of Hadoop, Starfish can give outstanding grades autonomously. Users are not required, therefore, to grasp and control the many Hadoops tuning knobs. However, we highlight how new data analysis procedures over huge data offer new issues, causing us to make various design decisions in Starfish based on self-tuning database systems.[6]

Chen, L. et al. [2018] The data in the storage system is maintained in a fine-grained manner thanks to a new replication strategy based on data blocks. Cloud storage costs and performance are balanced by E2FS, which examines the data's characteristics before making a replication choice based on dynamic data replication. We've built an E2FS prototype and tested it against HDFS to see how well it performs. According to our tests, E2FS outperforms HDFS in flexibility as guaranteeing presentation for huge data implementations.[7]

Manogaran, G. et al. [2018] Cloud computing and fog computing may be securely integrated using the GC architecture presented. Along with critical service facilities and data categorization functions in this design, intelligence services are also included (Sensitive, Critical, and Normal). An additional system component is an algorithm using MapReduce to forecast the onset of cardiac disease. The suggested infrastructures and predictions method's efficacy are shown via the use of metrics such as throughput, sensitivities, accurateness, and the F-measure.[8]

Wang, J. et al. [2020] Using this service architecture, service users may access a variety of customized data processing techniques, as well as data analysis and visualization. Data collection and storage are discussed in this paper's first introduction to the basic Big Data service architecture and the technical processing foundation. Next, we'll talk about handling and analyzing big data based on various service needs, which may provide significant information for customers. Next, we'll go through the specifics of the big data-based cloud computing service system, which offers great outstanding options for storing, processing, & analyzing enormous amounts of information. Finally, we provide several illustrations of how Big-Data could be used in various industries. [9]

Lu, Z. et al. [2018] It is hard to store all IoT Big Data in a single location because of the network's extreme latency and capacity constraints. The "edge cloud" idea, which distributes varied processing and data analysis capabilities across several edge clouds for IoT data analytics challenges, seems promising. MapReduce is a useful tool for dealing with large volumes of information. It is not easy to estimate how MapReduce processors will operate in the context of data analysis. IoTDeM, a more generalized IoT Big Data model for forecasting MapReduce performances across many edge clouds, is proposed and evaluated in this paper. MapReduce tasks in Hadoop 2 can be predicted with more accuracy by IoTDeM than they can in Hadoop 1, even with different reduction amounts and cluster sizes.[10]

Guo, J. et al. [2022] Earth observation (EO) today faces a huge challenge relating to storing and analyzing enormous volumes of remote sensing information (RS). Computational resources are given in a distributed network for high-speed processing of large amounts of real-time (RS) datasets. It follows that HDFS is constructed on K8s nodes, which are also utilized to do computing. In a Spark on Kubernetes (K8s) cluster for processing RS data, we adopt the tile-oriented programming paradigm instead of the typical pixel-oriented or strip-oriented approach. All computations may be divided into several distributed parallel jobs by abstracting any size user-defined raster tile format for an RS raster layer.[11]

Latifian, A. [2022] There have been issues with big data for enterprises, the IT industry, and the scientific community. Cloud computing and distributed computing technologies can successfully solve the challenges provided by big data. In the last year, Cloud Computing and Big-Data have emerged as 2 key issues for IT service providers to address to provide customers with high-efficiency and competitive computing tools on demand. Businesses may benefit from this research, which tries to investigate how the cloud can be used as a tool for handling large data in many ways.[12]

Xiang, Z. [2022] Unstructured data abounds in social media networks. In this research, visual dynamic simulation models for unstructured social network data have been proposed to assure the stability of large unstructured data. Data from unprocessed social network data, estimates and approximates of perceptual data, and an LR model due to the temporal correlations are all utilized in this study. The original social network's data is then filtered via an advisable threshold to eliminate the effects of clatter. With the help of feature analysis, we can extract unstructured material from social networks and identify its visual properties. Also included in this study are Hadoop cluster implementation, HDFS data permanence and MapReduce extraction clusters for distributed computing. Unstructured data in social networks may also be seen using a dynamic simulation model created by the software. An experiment conducted by the University of California, Berkeley, found that this method increases the durability, effectiveness, and aesthetics of unstructured material on social networking sites.[13]

Zhang, T. et al. [2022] Big data and cloud computing have made data management more difficult. Various data management and storage frameworks have emerged throughout time, each with its own unique set of characteristics and functions. Although they are very efficient, they end up creating data silos in the process. Because no one framework can successfully meet the data management demands of many applications, it goes harder to move and operate in harmony with information. This issue may be solved by using multi-tiered (hierarchies) storage technologies. An HSS may be constructed by combining many storage structures into a single, massive storage pool. Additionally, it offers a wide range of benefits, including increased storage utilization, cost-effectiveness, and the use of various storage framework parts. Hierarchical storage technologies must comprise intellectual & self-sufficient procedures for information control based on the attributes of the various structures in order to maximize their advantages. [14]

JANSEN, M. [2022] It is vital to provide reliable traffic information in metropolitan areas to minimize vehicle transportation's harmful effects. Urban planners and politicians who depend on Intelligent Transportation Systems (ITS) are increasingly concerned about proactively traffic control (ITS). This thesis focuses on road traffic forecasting (TF) for ITS systems, which extends the current research. This research in Belgium focuses on on-board-unit (OBU) data for heavy-goods trucks (HGV) [15]

Saravanan, V et al. [2022] IoT is a prominent issue in industry and academia due to recent improvements in cyber-physical systems and technical breakthroughs. As IoT transforms our daily life, it is also a technological and scientific revolution. People's everyday routines are greatly impacted as a consequence of this. Many intelligent systems exist for home automation, transportation, health, safety, and monitoring. Because so many devices are linked to these networks, they generate a tremendous volume of data that must be analyzed. Because of this, storage capacity, processing, and analysis become problematic. Big data and its related issues are discussed in this chapter, which concentrates on the IoT network and different data analytics solutions for the sector. As a result, they demonstrate and examine the IoT's big data architecture paradigm, along with many future research questions. They look at smart health and smart transportation as an extension of the architectural philosophy. [16]

Kumar, S. et al. [2022] A misuse of the current state of the executive-information connection, invention, examination, and perception or forecasting is proposed in big data engineering for SCM. It has also been discussed how to use the security and protection features of a Big-Data infrastructure in a real Big-Data framework organization when it comes to Supply Chain Management. An additional amount of employment has been brought to the author's notice.[17]

Cavicchioli, R. et al. [2022] Many potential developments may build on the proposed approaches' foundation of obstruction tracking and identification at the edges and aggregating information hierarchy in the cloud for traffic density monitoring. For the Modena Automotive Smart Zone (MASZ), the testing is based on an actual use case (MASA). [18]

Manivannan, P. et al. [2022] Data acquired from network sources is being crunched by small and medium-sized firms in an effort to make sense of the information. Data growth patterns and information transformations are becoming more important. A new term, "big data," was invented to describe data sets that are far larger and more difficult to integrate into company systems. Relational databases, which need a great deal of data processing, are the primary focus of most research. AI may therefore be used to input and expand the NoSQL document database, depending on the data type, because of this. Documented MongoDB is recognized in this study as real-time access to data stored on multiple storage platforms for all organization sizes. Datasets from big data analytics were used to build a NoSQL-MongoDB architecture with data sharing interwoven with AI and machine learning at the system level. Using this model, SMBs are presented with a limited perspective of database administration.[19]

Belcastro, L. et al. [2022] Because of these considerations, we present in-depth investigations of the key parallel processing architectures (MapReduce, Workflow BSP, Message Passing, and SQL-like) and, using examples, create the most often used systems for Big Data analysis (e.g., Hadoop, Spark, and Storm). As the various systems are compared and contrasted, we'll focus on the most important features and the development and user communities behind each one. We can assist designers and developers in choosing the optimal programming solution for their requirements based on their talents, hardware availability, application domains and goals, and the aid supplied by the developmental community generally. [20]

Enfais, A. M. A., et al. [2018] Large data sets may be stored and processed in a distributed computing environment using the Java-based Hadoop framework, which is well-suited to massive amounts of data. It stores data in HDFS and processes that data using MapReduce. The Map-reduce-based framework is well-known for handling information workloads operating on able-to-share clusters. The primary goal of the MapReduce programming paradigm is to distribute the processing of a single task over several nodes. Hadoop is becoming the primary focus of scholars and businesses alike. This has led to the development of a number of scheduling algorithms during the last several decades. MapReduce scheduling challenges include locality, synchronization, and fairness, among others.[21]

Allam, S. [2018] The purpose of this article is to learn more about Hadoop Log Analysis tools and how they might aid in technological advancements. Maintaining the quality of large-scale computer clusters is critical as their use grows. This elucidates the significance of network monitoring and control. The Hadoop cluster may be properly managed using a variety of approaches [1]. Each node in a cluster receives and processes the required data from these tools. [22]

Djafri, L. [2021] Large data processing tools and methods are summarised in this chapter by critically evaluating their aims and methodology and their major approaches to addressing the issues associated with big data processing. Using healthcare, smart cities, genomic sequence annotation, and graph-based applications as case studies, we evaluate some basic big data applications and their influence on human well-being. We present a complete assessment and categorization of the research activities within each application area. [23]

Ozdil, U. E et al. [2021] Reduced expenses are sometimes a result of a move to pay-as-you-go or pay-per-use commercially conceived cloud Hadoop PaaS from IaaS. As a result, the end-users of managed Hadoop systems are unable to see the advantages of using them because of their black-box nature. The study aimed to better understand how managed Hadoop context utilizes resources. We used three experimental Hadoop-on-PaaS approaches straight out of the box and ran HiBench Benchmark Suite workloads tailored to Hadoop specifically [31]. We monitored the worker nodes' system resource use throughout the benchmarks. According to the findings, the identical property standards across cloud services do not ensure local performance outcomes nor consistent results inside the cloud service itself. Pre-configurations and designs of the managed systems are assumed to have a major impact on performance.[32]

Sbai, I. et al. [2020] While Spark's in-memory processing power & GA's iteration procedures may be used, his parallelism tries to take advantage of both. The assessment of fitness and genetic procedures was carried out in parallel. A decision assistance system for the DVRP has been built using the parallel S-GA. For Big Data optimization challenges, the studies reveal that our suggested architecture is better because of its ability to link components and deploy over the whole cluster.[25]

3. Methodology

3.1. In this section, a Brief Study of the Bigdata File Handling Mechanism is Implemented Using.

3.1.1. Mathematical Analysis

Assume an n-disk system with p1, p2, p3 and pn failure probabilities. A hard drive failure is an isolated occurrence. So, Pf for hard disc failure is as follows:

$$F_{\beta} = \beta_1 \times p_2 \times p_1 \times \ldots \times p \tag{1}$$

$$0 \leq p \leq 1 \tag{2}$$

So,

$$p_f \leq \forall_1 \tag{3}$$

This reduces the risk of hard disc failure. But if one of the hard discs fails, the system fails. So, the system's failure probability is I.

$$3F_{fx} = MAX\left(p_{1,}p_{2,}p_{8,} \ldots p_n \right) \tag{4}$$

Where P_{fss} is the system failure probability when discs are placed in series. So the chance of a system breakdown is

$$4\; P_f \geq \; P_{i,} i = \; 1, 2, 3 \ldots n \tag{5}$$

The quantity of hard drives in a system raises the likelihood of system failure.

II. The larger the hard disk's capacity, the longer it takes to retrieve data.

3.1.2. Mathematical Evaluation Analysis

Assume the maximum hard disc search time is 0.4 ms and the chance of data availability is 0.5ms. So, total seek time in -system serial hard disc is performed high time of conversion functions. Hadoop has improved the file handling process in this work using mathematical computations.

$$T_1 = \{(a_1 \times s_1) + [1 - p_{a1}]\} \times p_a \;\times s_2 \;+ \cdots + \\ (1 - p_{a1}) \;\times\; (1 - p_a) \times \ldots \times (1 - p_{an-1}) s_{u-1} \tag{6}$$

So overall search time goes up. So, adding another hard drive appears impracticable. Even while the mathematical study shows that adding extra hard discs slows down the system, it does provide one ray of hope. Individual hard disc failure probability is smaller than overall failure probability (see equation 1). So, adding a hard disc is beneficial. Hard disc failure rate drops.

3.1.3. Adding Hard Disk in Parallel

We may use the result [eqn-3] in some way. In this case, the failure of one hard drive does not affect the total system. In such instances, system failure probability is independent of hard disc failure [27]. The system will die if and only if all hard drives fail.

$$p_{f \to 1} = p_1 \times \forall_1 \;\times\; t_1 \;\times \ldots \times\; \forall_2 \tag{7}$$

P_{fsp} is the likelihood of system failure when adding parallel hard drives. Comparing equations (4) and (7), we can see that adding discs in parallel reduces the total likelihood of system failure. Parallelizing hard discs improves performance in terms of disc failure, but it also helps us in terms of speed. How a file is sought on one hard drive might help us solve this issue.

The file is saved like this:

A node and a directory hold the file's content and metadata. Metadata is searched first rather than data blocks whenever a search is performed.

We can take metadata from each hard drive and store it on a single hard drive. A single hard drive search is required in this case. In this case, the seek time is as follows:

$$T_Y = p_{d1} \times S_1 \tag{8}$$

So the search time is lowered [compare equations (6) and (8)].

3.1.4. General Architecture

Based on the aforementioned, a tree-based architecture is presented [Figure. 3].

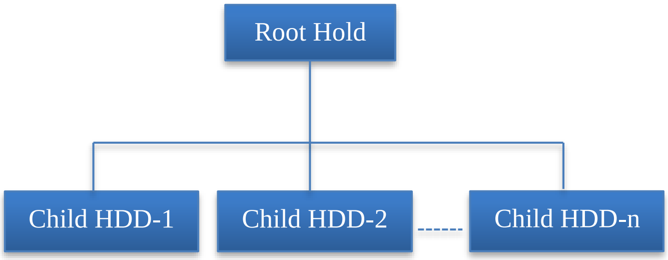


Fig. 3 Hierarchical structure of hard disk

The root of the tree stores metadata, while other nodes store data. This design decreases search time and thereby reduces system failure. Now the system fails if all children fail [6].

3.2. Pros and Cons

The root holds all metadata.

- Because metadata is tiny, root storage is not needed.
- Faster data searching. This requires a quicker algorithm.
- Root's processing power must be high since several users may need data at once.
- Data is lost if the root node fails.
- Storage capacity must be big since the child node only stores data.
- The kid is not only storing and sending info.
- In order to store and retrieve data, the child hard disk's block size must be big enough. This reduces traversal time [28].

3.2.1. Concept of Distributed File System

Mathematics proposes a very obvious design where data is stored on several hard discs. The data is stored or dispersed over a network of hard discs. This design overcomes most storage issues and necessitates a storage management system [7,8]. The DFS does the following:

- Selecting node data storage.
- What if the data storage fails?
- How to retrieve data if a node fails?
- How to save et info for quick access?

3.2.2. Concept of Hadoop

This section describes Hadoop's architecture. Hadoop has a master/slave design for both distributed storage and computing. MapReduce is a distributed computing framework that uses Hadoop Distributed File System (HDFS). So Hadoop has two important parts [29].

- Hadoop Distributed File System
- MapReduce (Processing)

The above modeling can finalize the storage space size as well as allocate the proper vector location. Based on child node roots, they have processed the data. The leaf node is the final output-carrying node that can manage the MapReduce process [30]. The Hadoop model has continually monitored data in various steps using clustering and unstructured data handling. The root node can deliver instructions to the child and leaf nodes based on the steps allotted, and file storage has been functioning. The Hadoop Distributed File System with the proposed technique achieved more improvement in the file storage process shown in Figure 4.

Hadoop File Systems

Hadoop is a file system developed for huge file storage and streaming data access on commodity hardware clusters. AHDFS comprises two parts.

a) Name Node

The system's master maintains and handles blocks on DataNodes. In addition, it monitors the distributed file system's general health and how your data are divided down into file blocks—a Hadoop cluster's single point of failure.

b) Data Nodes

These are the storage nodes that are placed on each system. 10

c) Secondary Name Node

It is a monitoring method for the Hadoop cluster. Because a Hadoop cluster's NameNode is a single point of failure, the secondary NameNode helps reduce downtime and data loss.

MapReduce

MapReduce is the Hadoop system's beauty. It is a programming approach for processing huge data sets on a cluster in parallel. MapReduce is a combination of two terms: Map and Reduce. In the first case, a data collection is converted into another data set, with each piece broken down into tuples (key/value pairs). The reduction task takes a map's output and compresses it into a smaller collection of tuples. As the term MapReduce indicates, the reduction task always comes after the map job.

The master has a Job Tracker, and each enslaved person has a Task Tracker for MapReduce. The Task Tracker daemon monitors a job across several nodes. If a node fails, the Job Tracker reschedule the job. Task Tracker keeps track of the tasks allocated to each node shown in Figure 5.

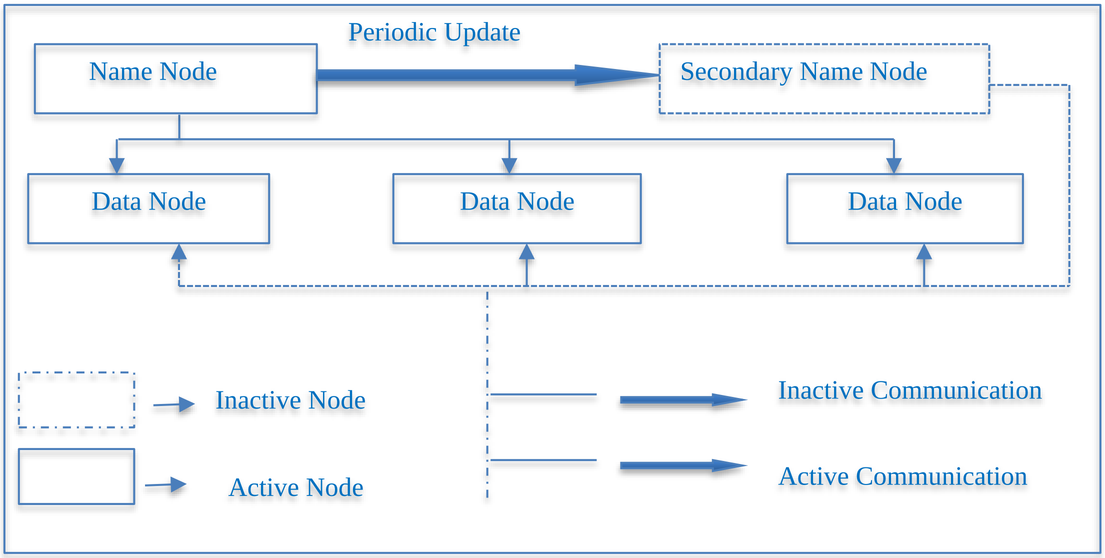


Fig. 4 Hadoop file system architecture

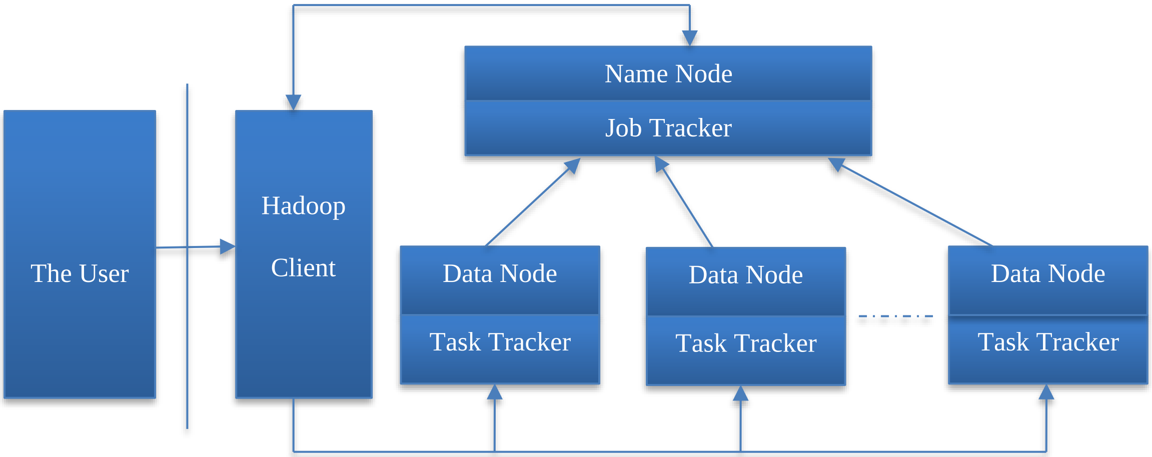
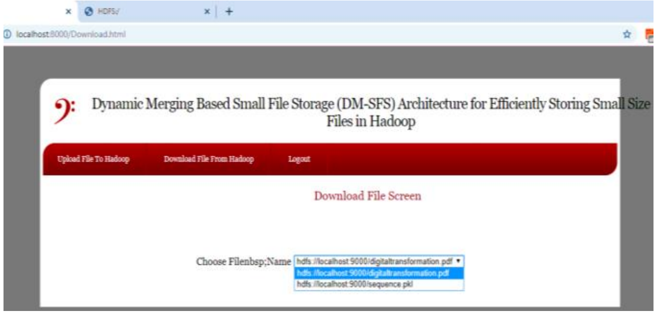


Fig. 5 The Hadoop System

4. Results and Discussions



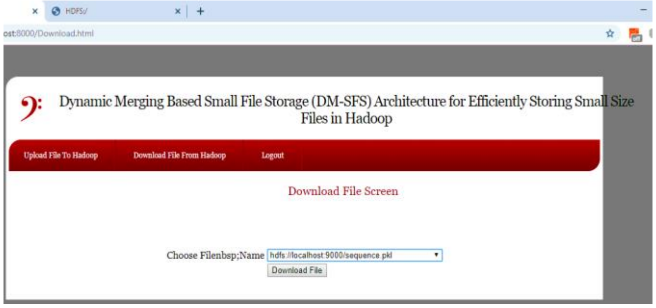
localhost:8000/Download.html

Dynamic Merging Based Small File Storage (DM-SFS) Architecture for Efficiently Storing Small Size Files in Hadoop

Download File Screen

Choose Filenbsp;Name hdfs://localhost:9000/digitaltransformation.pdf
hdfs://localhost:9000/digitaltransformation.pdf
hdfs://localhost:9000/sequence.pkl

Fig. 6 Hadoop-based file handlining screen



ost:8000/Download.html

Dynamic Merging Based Small File Storage (DM-SFS) Architecture for Efficiently Storing Small Size Files in Hadoop

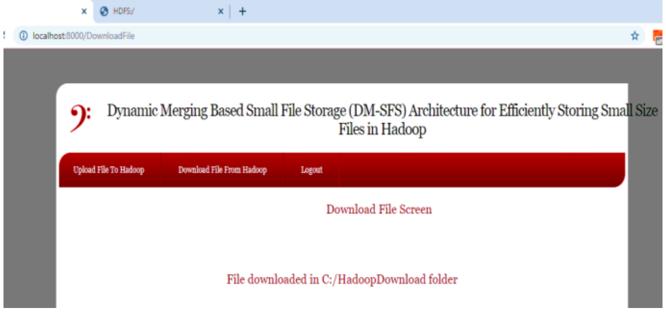
Download File Screen

Choose Filenbsp;Name hdfs://localhost:9000/sequence.pkl

Fig. 7 Downloaded file

All uploaded files are displayed in the drop-down box in the above screen, and we can select any file to download to the 'C:/Hadoop Download' folder illustrated in Figure 6.

In this picking sequence in the above file, use the Hadoop process. The processed file will be downloaded in the C:/HadoopDownload folder, containing a merge of two files.



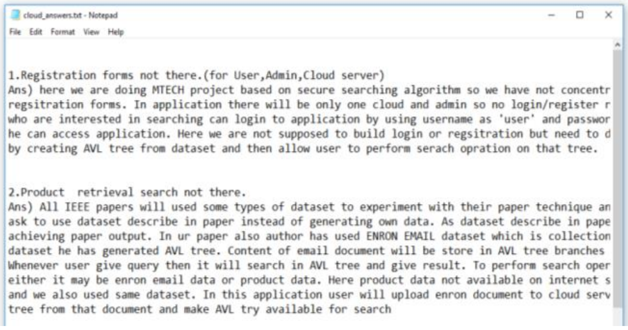
Dynamic Merging Based Small File Storage (DM-SFS) Architecture for Efficiently Storing Small Size Files in Hadoop

Download File Screen

File downloaded in C:/HadoopDownload folder

Fig. 8 Location folder

In this process, we can see the file downloaded in the C:/HadoopDownload folder on the previous screen, and we will now examine that folder shown in Figure 9.

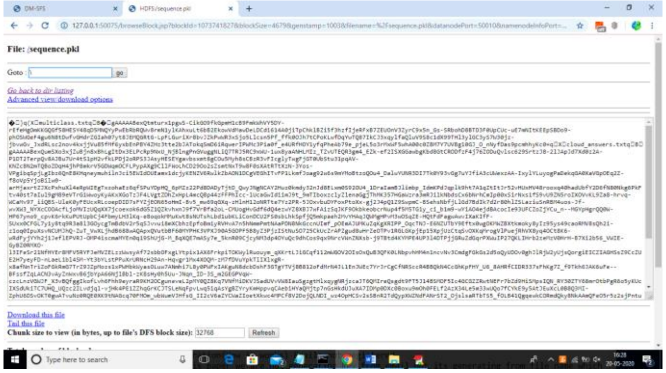


```
1.Registration forms not there.(for User,Admin,Cloud server)
Ans) here we are doing MTECH project based on secure searching algorithm so we have not concentr
regsitration forms. In application there will be only one cloud and admin so no login/register r
who are interested in searching can login to application by using username as 'user' and passwor
he can access application. Here we are not supposed to build login or regsitration but need to d
by creating AVL tree from dataset and then allow user to perform serach opration on that tree.


2.Product  retrieval search not there.
Ans) All IEEE papers will used some types of dataset to experiment with their paper technique an
ask to use dataset describe in paper instead of generating own data. As dataset describe in pape
achieving paper output. In ur paper also author has used ENRON EMAIL dataset which is collection
dataset he has generated AVL tree. Content of email document will be store in AVL tree branches
Whenever user give query then it will search in AVL tree and give result. To perform search oper
either it may be enron email data or product data. Here product data not available on internet s
and we also used same dataset. In this application user will upload enron document to cloud serv
tree from that document and make AVL try available for search
```

Fig. 9 Merge file

The whole sequence explains that all data merging files have been successfully transferred and decoded, as shown in the screenshot above. In the screenshot below, the same sequence file is visible in an encrypted format on the Hadoop server, shown in Figure 9.


DM-SFS
HDFS:/sequence.pkl
127.0.0.1:50075/browseBlock.jsp?blockId=1073741827&blockSize=4679&genstamp=1003&filename=%2Fsequence.pkl&datanodePort=50010&namenodeInfoPort=...
go
Refresh
Type here to search
16:28
20-05-2020

File: /sequence.pkl

Goto :

Go back to dir listing
Advanced view/download options

```
[illegible]
```

Download this file
Tail this file
Chunk size to view (in bytes, up to file's DFS block size): 32768

Fig. 10 encrypted format

The sequence content is shown in the upper screen from the Hadoop server, which was in an encrypted format, as seen in Figure 10, which describes the file's encryption format.

The table 1 clearly explains various existing models and proposed model comparisons. In this work, the proposed ENR-based file handling model attains more improvement.

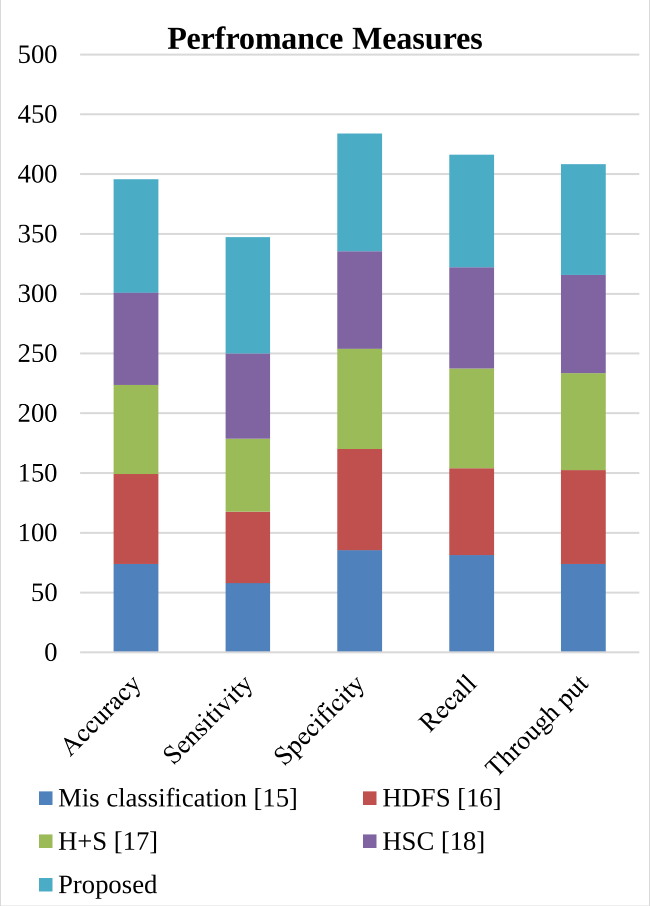


Fig. 11 Comparison of results

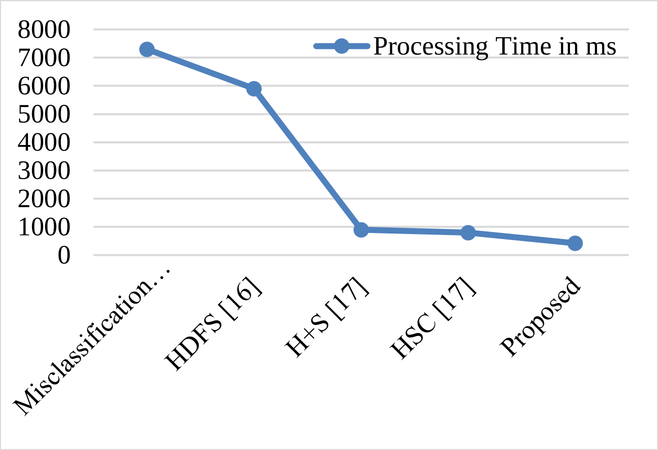


Fig. 12 Processing time

Table 1. Comparison of results

Methods	Accuracy	Sensitivity	Specificity	Recall	Throughput
Misclassification [15]	74.22	57.67	85.39	81.42	74.2
HDFS [16]	74.84	60.07	84.77	72.58	78.1
H+S [17]	74.76	61.00	84.02	83.45	81.2
HSC [17]	77.032	71.230	81.340	84.78	82.1
Proposed	94.87	97.42	98.60	94.06	92.7

Table 2. Processing time

Processing Time	
Methods	**Processing Time in ms**
Misclassification [15]	7300
HDFS [16]	5900
H+S [17]	900
HSC [17]	800
Proposed	420

Figure 11 and Table 2 describe all existing performance metrics and the suggested approach MSR+ENR, which clearly explains how the proposed method provides more improvement.

The processing time allocation to each approach is explained in Figure 24 and Table 2, and the suggested method MSR+ENR achieves a bigger improvement than the existing methods shown in Figure 12.

5. Conclusion

This research study used HDFS+ENR+MSR, a novel file-handling technique. It is a contemporary solution that addresses the Name-node and Data-node issues associated with files. The purpose is to solve the issue identified by the survey using the Java-8u-121 extension and Python 3.7 software. This strategy is successfully handling files of all sizes, both huge and tiny. The data storage strategy is primarily split into two categories: if the selected file is greater than 100 kB, it is automatically recognized as a tiny file and stored in the general area; otherwise, it is saved to the Hadoop server using the merging option. The suggested MSR+ENR provides a better improvement in this study when compared to existing techniques. Accuracy levels of roughly 95%, sensitivity levels of 98%, specificity levels of 99%, recall levels of 95%, throughput levels of 92%, and processing times of 420 ms are reached; these findings challenge state-of-the-art technology.

References

[1] Hung-Ming Chen, Kai-Chuan Chang, and Tsung-Hsi Lin, “A Cloud-Based System Framework for Performing Online Viewing, Storage, and Analysis on Big Data of Massive BIMs,” *Automation in Construction*, vol. 71, pp. 34-48, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[2] Mohd Abdul Ahad, and Ranjit Biswas, “Dynamic Merging based Small File Storage (DM-SFS) Architecture for Efficiently Storing Small Size Files in Hadoop,” *Procedia Computer Science*, vol. 132, pp. 1626-1635, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[3] Yongqiang He et al., “RCFile: A Fast and Space-Efficient Data Placement Structure in MapReduce-based Warehouse Systems,” *IEEE 27th International Conference on Data Engineering*, pp. 1199-1208, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[4] Mohd Rehan Ghazi, and Durgaprasad Gangodkar, “Hadoop, MapReduce and HDFS: A Developers Perspective,” *Procedia Computer Science*, vol. 48, pp. 45-50, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[5] K. R. Krish et al., “On Efficient Hierarchical Storage for Big Data Processing,” *16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, IEEE*, pp. 403-408, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[6] Herodotos Herodotou et al., “Starfish: A Self-tuning System for Big Data Analytics,” *CIDR*, vol. 11, no. 2011, pp. 261-272, 2011. [Google Scholar] [Publisher Link]

[7] Longbin Chen et al., “E2FS: An Elastic Storage System for Cloud Computing,” *The Journal of Supercomputing*, vol. 74, no. 3, pp. 1045-1060, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[8] Gunasekaran Manogaran et al., “A New Architecture of Internet of Things and Big Data Ecosystem for Secured Smart Healthcare Monitoring and Alerting System,” *Future Generation Computer Systems*, vol. 82, pp. 375-387, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[9] Jin Wang et al., “Big Data Service Architecture: A Survey,” *Journal of Internet Technology*, vol. 21, no. 2, pp. 393-405, 2020. [Google Scholar] [Publisher Link]

[10] Zhihui Lu et al., “IoTDeM: An IoT Big Data-Oriented MapReduce Performance Prediction Extended Model in Multiple Edge Clouds,” *Journal of Parallel and Distributed Computing*, vol. 118, pp. 316-327, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[11] Jifu Guo, Chunlin Huang, and Jinliang Hou, “A Scalable Computing Resources System for Remote Sensing Big Data Processing Using GeoPySpark Based on Spark on K8s,” *Remote Sensing*, vol. 14, no. 3, p. 521, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[12] Ahmad Latifian, “How Does Cloud Computing Help Businesses to Manage Big Data Issues,” *Kybernetes*, vol. 51, no. 6, pp. 1917-1948, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[13] Zhang Xiang, “Visual Dynamic Simulation Model of Unstructured Data in Social Networks*,”* *Security and Communication Networks*, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[14] Tianru Zhang, Salman Toor, and Andreas Hellander, “Efficient Hierarchical Storage Management Framework Empowered by Reinforcement Learning,” *Arxiv Preprint arXiv:2201.11668*, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[15] Maarten Jansen, “*On-Board-Unit Big Data Analytics: from Data Architecture to Traffic Forecasting*,” Doctoral Dissertation, Katholieke Universiteit Leuven, 2022. [Google Scholar] [Publisher Link]

[16] Vijayalakshmi Saravanan, Fatima Hussain, and Naik Kshirasagar, “Role of Big Data in Internet of Things Networks,” *Research Anthology on Big Data Analytics, Architectures, and Applications, IGI Global*, pp. 336-363, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[17] Sundeep Kumar, Vikram Singh Rathore, and Alok Mathur, “An Analytical Study on Big Data Management for Supply Chain Analytics,” *Recent Advances in Industrial Production*, Springer, Singapore, pp. 333-341, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Roberto Cavicchioli, Riccardo Martoglia, and Micaela Verucchi, “A Novel Real-Time Edge-Cloud Big Data Management and Analytics Framework for Smart Cities,” *Journal of Universal Computer Science*, pp. 3-26, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[19] P. Manivannan, D. Prabha, and K. Balasubramanian, “Artificial Intelligence Databases: Turn-on Big Data of the SMBs,” *International Journal of Business Information Systems*, vol. 39, no. 1, pp. 1-16, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[20] Loris Belcastro et al., “Programming Big Data Analysis: Principles and Solutions,” *Journal of Big Data*, vol. 9, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[21] Abolgasem M. Ali Enfais et al., “Enhancing Hadoop Performance in Homogeneous Big Data Environment Assuming Configuration of Dynamic Slots in Map-Reduce Pattern,” *International Journal of Engineering & Technology*, vol. 7, no. 4, pp. 6986-6990, 2018. [Google Scholar] [Publisher Link]

[22] Sudhir Allam, “An Exploratory Survey of Hadoop Log Analysis Tools," *International Journal of Creative Research Thoughts*, vol. 6, no. 8, pp. 801-804, 2018. [Google Scholar] [Publisher Link]

[23] Laouni Djafri, “Dynamic Distributed and Parallel Machine Learning Algorithms for Big Data Mining Processing,” *Data Technologies and Applications*, vol. 56, no. 4, pp. 558-601, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[24] H K Pradeep, K Rohitaksha, and C B Abhilash, "An Email based Offline Download Manager for Large Distributed File System using Hadoop MapReduce Framework," *SSRG International Journal of Computer Science and Engineering*, vol. 1, no. 10, pp. 1-5, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[25] Ines Sbai, and Saoussen Krichen, “A Real-Time Decision Support System for Big Data Analytic: A Case of Dynamic Vehicle Routing Problems,” *Procedia Computer Science*, vol. 176, pp. 938-947, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[26] Bunmi Deborah Millennial-Oriagbo, and Muhammad Ghali Aliyu, "Pros and Cons of Big data in a Global Digital Transformation," *SSRG International Journal of Mobile Computing and Application*, vol. 8, no. 3, pp. 1-10, 2021. [CrossRef] [Publisher Link]

[27] O Sai Saran et al., “3D Printing of Composite Materials: A Short Review,” *Materials Today: Proceedings*, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[28] Kiran Dasari, Lokam Anjaneyulu, and Jayaraju Nadimikeri, “Application of C-Band Sentinel-1A SAR Data as Proxies for Detecting Oil Spills of Chennai, East Coast of India,” *Marine Pollution Bulletin*, vol. 174, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[29] K. Sarada et al., “Records of Patient Health Data and Medical Information Monitoring Using IOT,” *2nd International Conference for Innovation in Technology, IEEE,* pp. 1-6, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[30] S. Murugan et al., “Impact of Internet of Health Things (IoHT) on COVID-19 Disease Detection and Its Treatment Using Single Hidden Layer Feed Forward Neural Networks (SIFN),” *How COVID-19 is Accelerating the Digital Revolution: Challenges and Opportunities, Cham: Springer International Publishing*, pp. 31-50, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[31] Chandra Shaker Pittala, Vallabhuni Vijay, and B. Naresh Kumar Reddy, “1-Bit FinFET Carry Cells for Low Voltage High-Speed Digital Signal Processing Applications,” *Silicon*, vol. 15, pp. 713-724, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[32] Uluer Emre Özdil, and Serkan Ayvaz, “An Experimental and Comparative Benchmark Study Examining Resource Utilization in Managed Hadoop Context,” *Cluster Computing*, vol. 26, pp. 1891-1915, 2021. [CrossRef] [Google Scholar] [Publisher Link]