K-means Clustering with AES in Hadoop MapReduce

Chunmei Ji[1], Jing Zhou [2*], Fengbo Kong[2]
[1]Department of Information & Security, Yancheng Polytechnic College, Yancheng, Jiangsu 224005, China
[2] Yunnan Industry & Trade Vocational College, Kunming, Yunnan 650300, China
E-mail: ttdsxqt20230222@163.com, 1995170164@yctei.edu.cn
*Corresponding author

Keywords: distributed computing, data management, prediction strength, cloud environment, clustering algorithm, aes encryption method, hadoop

Recieved: April 19, 2024

The goal of this study is to find the best way to employ parallel processing to optimize the clustering method according to the anticipated intensity. According to the experimental findings, picture clustering becomes stable if the total amount of clusters above a certain threshold, suggesting that it becomes less susceptible to random influences. Using a method of optimization revealed that four clusters provided the most accurate cluster proof of identity, even if it was not able to pinpoint the exact best number of clusters. The ideal number of clusters was established by doing extensive testing. Clustering analysis reveals that most columns get more attention from visitors who exhibit certain characteristics. With this approach, big datasets with missing or partial data may be more easily clustered, and clustering speed and accuracy are both enhanced. This technique optimizes clustering based on predictions strength in the cloud, which improves processing precision as well as effectiveness in large amounts of data when compared to previous approaches.

Povzetek: Raziskava uvaja optimizacijo algoritma za razvrščanje, ki uporablja AES šifriranje v Hadoop MapReduce okolju za izboljšano obdelavo podatkov. Ta pristop omogoča natančno klasifikacijo velikih podatkovnih nizov v oblaku z večjo hitrostjo in točnostjo, kar povečuje učinkovitost obdelave pri obsežnih podatkih.

1 Introduction

One of the most important fuzzy clustering tools in data mining and recognition of patterns, the possibilistic c-means approach (PCM) finds extensive use in image analysis as well as knowledge discovery. However, it may be challenging to get acceptable clustering results for large data, particularly when the data is varied, using PCM, since it was initially designed for small structured data sets [1]. To enhance the capability of mathematical attribute mining to exchange big data, it is essential to optimize clustering processes of numerical attribute information about features in big data. Consequently, a cloud-based technique for clustering big data was suggested. We retrieved the amount of the mutual information feature from the numerical attributes of the communication big data using the cloud extended distributed features fitting approach, which was applied to the numerical attributes of the data using linear programming [2]. Streaming media placement and caching are best handled by cloud-edge cooperation architectures, which integrate processing at the edge with processing in the cloud. The four sub-problems that make up the model are as follows: job categorization, node clustering, node resource allocation, and cache-aware task scheduling. First, there is a three-part classification system for tasks; second, nodes get resources based on the circumstances under which their respective activities are executed [3]. The problem of protecting and managing large amounts of data stored in the cloud has grown in recent years. For secure and trouble-free cloud computing, it is important to use best practices, methodologies, and approaches. At the same time, data analytics methods are crucial for making use of huge data [4]. The fact that K-means clustering only merges to adjacent optimums—which is easier than understanding for global optimums but might lead to less optimal union—is one of its fundamental difficulties. This is particularly true for massive datasets, as the underlying foci have a major role in the presentation of this computation [5]. When thinking about cloud computing, work scheduling is a difficult problem, particularly when considering deadlines and costs. However, achieving efficient allocation of user jobs in clouds is the central challenge of task scheduling. The execution of a large number of jobs using a sequential method increases the computational cost due to memory

time and space complexity [6]. Data processing speeds will need to rise to keep up with the ever-increasing creation and distribution of big data from all kinds of sources. The job of the task scheduler in cloud computing and other distributed big data processing platforms is to allocate a wide variety of jobs across a group of potentially diverse computer nodes in a manner that maximizes data locality and efficiency while minimizing makespan [7]. State data in a smart grid setting is characterized by being comprehensive, massive, and dependable. Database management software makes use of relational database systems, whereas conventional storage hardware use disk arrays. It is difficult to adapt to needs since the system is not scalable, expensive, or reliable enough [8]. Many people are starting to pay attention to the idea of using cloud-based resources and high-performance computing to handle massive amounts of medical imaging data. Microwave imaging is being explored for use in brain and breast scans since it is non-invasive and inexpensive. Due to the great quality of the pictures, it produces, microwave imaging using the space-time method and its expanded variants sees widespread application [9]. A growing number of enterprises are using Hadoop, a distributed computing system capable of quickly processing large-scale information, as the foundational computing framework for their cloud computing platforms. There is a lot of interest in studying how to make Hadoop run more efficiently, and one of the main factors influencing this is the scheduling issue. It is critical to find its flaws and fix them [10]. Big data analytics is a methodical approach to sifting through massive datasets in search of trends, patterns, and correlations. Processing and analyzing big data require a lot of time, energy, and specialized tools. For rapid, large-scale data analysis, the Hadoop platform's software employs the MapReduce method, which makes advantage of parallel processing [11]. Most companies nowadays employ cloud computing to handle large data by making advantage of cloud resources and services. Furthermore, the most useful information is extracted from raw data supplied by various sources using machine learning algorithms [12]. The quantity of data stored online is expanding at an exponential rate in this age of big data. The problem of how to rapidly extract useful information from large datasets has grown in complexity.

A customized e-commerce recommendation algorithm built on Hadoop is aimed at solving the issues in the collaborative filtering recommendation method [13]. This algorithm will help with data sparsity, scalability, with real-time recommendation, all of which are challenges in recommendation technology. Recent years have seen an explosion of data types, both permanent and transient, which has put pressure on big data processing platforms to increase their processing speeds. Using as many parallel processing as needed to handle data quickly is one limitation of operating such platforms in cloud computing settings. As a result, nodes in a cloud setting often face very congested groups of CPUs. In order to process data quickly with a high degree of parallelism, effective systems for allocating tasks and resources are necessary [14]. Storage and administration of large data have become more important as a result of new technology like the Internet and cloud computing. Concurrently, cloud storage systems face new demands from emerging cloud applications, including high concurrency and great scalability. To allow the dynamic growth and demise of virtual nodes, the current nosql database architecture is built on cloud computing virtual resources [15].

Using an online learning platform and a large data cloud computing platforms as an example, we break down the system's functional modules according to the user's needs, and then we build the system in a nutshell to figure out its architecture. Online classroom, online experiment, online assessment, video course, and basic function modules make up the system's key functionalities [16]. The Apriori algorithm, which is tasked with mining collections of frequently used items, is one of the best-known and most-used data mining algorithms. Many new algorithms, implemented on distributed and parallel platforms, have recently enhanced the Apriori algorithm's performance. The data architecture, manner of data degradation, memory system, and load balancing technique that were used to build them make them different from one another [17]. A major change from the previous paradigm of computer capacity acquisition—ownership—to the current subscription strategy is represented by cloud computing. Researchers are encouraged to delve further into the advantages of cloud resources for running scientific applications like workflows using cloud computing, which involves the provisioning and deployment of services in a distributed environment. Some workflow activities are more fine-grained, while others are more coarse-grained [18].

1.1 Distributed computing in big data - cloud

A great deal of recent technological development has allowed for an explosion of data, thanks to a plethora of new innovations. The process of coordinating the efforts of several computers to address a shared issue is known as distributed computing. It allows many computers to operate together as if they were a single supercomputer, allowing for massive resource allocation to tackle difficult problems. The most typical instance of a massive amount of data is the history of internet searches. A large number of individuals all across the globe will look for the data they need. It becomes a very difficult problem to manage this massive

amount of data. Hadoop was born out of the need to handle such massive amounts of data. Distributed data storage and parallel processing are both made possible by Hadoop. Hadoop clusters fundamentally use a master-slave paradigm for data storage. Hadoop is an open-source software framework that facilitates the development and execution of applications capable of handling massive data sets. Hadoop, which is available under the Apache Free Software License v2.0, was first developed to facilitate widespread file searches. Hadoop efficiently supplies the big data infrastructure and was created to handle the many problems associated with large data. Hadoop is primarily categorized into two primary mechanisms:

(a). Processing (map reduce)

To analyze and generate massive data sets using a distributed, parallel method on a cluster, Map Reduce provides a software design paradigm and an accompanying implementation. An example of a Map Reduce program would be sorting students into queues according to their first names and then filtering that data. Then, the program would have a Reduce method that would execute an immediate operation, like counting the total amount of students in each queue to get their name frequencies. When it comes to processing, the Map Reduce System (also known as a framework or structure) scores high marks for its ability to organize distributed servers, run numerous tasks in parallel, manage transportation and information transfers between components, and ensure redundancy and fault tolerance. Though their initial intent is different in the Map Reduce framework, the paradigm is inspired by the map and reduce functions that are popular in functional programming. Not the mapping and reduce operates themselves, which are similar to the reduce and disperse operations in the Message Passing Interface standard, but the ability to scale and load tolerance attained for a variety of applications through a single improvement to the operation of the engine, are the primary drivers of the Map Reduce framework. Therefore, it is common for multi-threaded implementations of Map Reduce to exhibit significant performance gains, but single-threaded implementations of the algorithm will often not outperform their non-Map Reduce counterparts. This paradigm is useful only when the Map Reduce framework's fault tolerance features and better dispersed shuffle operation, which lowers the cost of network transmission, are in use. To be effective, a Map Reduce algorithm must reduce the cost of transmission.

Many different programming languages have produced Map Reduce libraries, yet their optimization levels couldn't be more different. Apache Hadoop is a well-liked open-source implementation that includes support for distributed shuffles. The term Map Reduce has subsequently extended beyond its initial association with Google's exclusive technology. Apache Mahout's research has shifted away from disk-oriented devices with more capabilities and toward devices with complete map and reduce capabilities, therefore Google stopped utilizing Map Reduce as their main approach for processing big data. By using the jetty server for Hadoop's map phase, our suggested solution efficiently retrieves data in the cloud using the map-reduce paradigm, which in turn reduces the total time it takes to get data. The Map Reduce programming approach consists of two phases: the Map phased phase and the Reduce phase. It breaks the files into many smaller pieces of work. After that, HDFS makes sure the data blocks are reliable by creating copies and storing them on computing nodes throughout the cluster. At last, Map Reduce can handle data locally.

- **Map phase**: The input data and key quantity pair are sent through to it, and it then emits a pair of intermediate keys.
- **Reduce phase**: Combines the input key and value from the intermediary with the same key to produce the resultant key and value.

(b). Storage (hadoop distributed file system)

HDFS is in charge of storing and managing massive amounts of data that will be handled by the processing component. The primary node in an HDFS cluster is responsible for managing user permissions on files and the file systems space. All nodes that operate on data may have their storage managed by data nodes (Slave). Users may access the data nodes that house their files using a name node. on order to store files on data nodes, they might be broken into many pieces. The following objectives are intended to be accomplished by the HDFS: The organization and storage of datasets is be a challenging task to manage. Applications that handle with enormous data sets are managed via HDFS. For HDFS to do this, every cluster has to have hundreds of nodes.

- Fault detection—Given the amount of commodity hardware included in HDFS, there must be technology to search for and identify defects efficiently. Component failure is a typical problem.
- Hardware efficiency — Using it with big datasets may boost processing performance and decrease traffic through the network.

1.2 Contributions

Data security has become an important concern in the age of Big Data. Here are the main points of this paper:

- In order to safeguard the data stored in Hadoop against unknown users, this study employs the Advanced Encryption Standard, also known as AES encryption. Those who have permission to access the file are the only ones who can open it. The Data Node encryption algorithm was subjected to extensive testing. Hadoop's encryption technique proved its worth by securely encrypting the file's contents.
- Applying the serial k-means clustering approach to massive datasets with sizes in the tera and petabyte range requires a significant amount of processing power, which is why the MapReduce architecture is suggested.
- After that, a MapReduce model that uses K-means clustering to group the encrypted data is introduced. You may save this information in the HDFS database. The contemporaneous K-mean clustering approach makes use of a MapReduce technology that is based on the sharing of information concept. When two processes that are running in parallel share data, it's called a synchronous interaction.
- The proposed parallel clustering utilizing k-means algorithm much surpasses the serial k-means approach in experimental results, demonstrating fault-tolerant and parallel execution. Results from experiments demonstrate that, particularly on multi-core CPUs, the parallel technique drastically cuts down on processing time. Similarly, the estimated plot yields adequate results with little runtime.

1.3 Paper structure

Here is how the rest of the article is structured. Part II provides an overview of the most important relevant studies. The suggested model, down to the most basic ideas, is laid forth in Section III. The suggested model's safety and efficiency analysis, as well as a short comparison with current systems, are presented in Section IV. Section V concludes the report by suggesting avenues for further research.

2 Related work

To get around this issue, researchers in [19] suggest a high-order PCM method for clustering huge data sets. This method optimizes the objective function utilizing tensor space. Additionally, they use MapReduce to construct a distributed HOPCM approach that can handle very big and diverse datasets. To safeguard sensitive data stored on cloud servers, authors develop a privacy-preserving HOPCM algorithm that employs the BGV encryption approach. To make the BGV method computations safer, PPHOPCM utilizes polynomial approximations for the functions that update the membership matrix and clustering canters. Evidence from trials suggests that PPHOPCM can securely cluster massive amounts of heterogeneous data in the cloud without disclosing any personally identifiable information.

Reviewing a paper on clustering methods in the cloud for the purpose of analyzing and clustering massive datasets is the subject of [20]. The research delves into the process of applying and assessing different clustering algorithms inside a cloud-based computing setting. It stresses the need of using cloud resources to effectively manage enormous amounts of data and discusses the challenge of big data clustering. Using real-world datasets, the authors demonstrate experimental findings and suggest a new method that conducts distributed clustering on cloud infrastructure. Cloud computing and big data clustering are the subjects of this paper's extensive research study. Discussing pertinent past research, it draws attention to the gaps and issues that the proposed strategy aims to overcome.

Using the idea of distributed clustering in a cloud computing setting, the authors of the [21] article suggest an enhanced K-means clustering method. Combining the enhanced method (T.K-means) with the Hadoop platform's MapReduce computing architecture enables parallel computation, allowing for the analysis of enormous data sets. Using milling data of Ti-6Al-4V alloy as the mining object, they can test the practical performance of the T.K-means method. The optimal value for the cutting parameters is derived by mining the mapping connection between the parameters, roughness of the surface, and material removal rate. In order to get the optimum surface roughness while milling Ti-6Al-4V titanium alloy, the findings demonstrate that the T.K-means method may be used to mine the ideal cutting parameters.

For a large-scale distributed power system's big dataset [22] for an analysis of the cloud computing environment's Hadoop distributed platform. Common data mining method modules, like the parallel real-time clustering method and the parallel classification algorithm, are developed in accordance with the properties of intelligent electrical consumption data, and the application of clustering algorithms based on different principles is subsequently examined. A distributed database system called HBase is then deployed, and MapReduce can be utilized to make power GIS data visual administration possible. The algorithm has strong identification capabilities for large datasets in cluster mode, and the experimental findings demonstrate that the Hadoop-based parallel processing approach to power big data is both efficient and scalable.

A new K-means clustering approach is suggested in [23], which provides a method to find advanced areas of initial foci and an initial number of bunches. This promotes fast and accurate grouping across massive datasets, and it leads to obtaining the final arrangement of bunches to meet worldwide. Distributed computing allows for the execution of sophisticated and expansive processing tasks. The use of the parallelism approach allows for the efficient and cost-effective breakdown of large amounts of data. By combining R Studio's Flexible Process Cloud event with Amazon Web Services, they can achieve parallelism and flexible registration by dividing the work across many hubs. In a fraction of the time and with much less expense, the suggested method puts on a very serious show.

To address the scheduling issue with huge data in the cloud, the authors of this study provide a solution based on antagonistic GWO for efficient memory constraint parallelized resource allocation and optimum scheduling [24]. Using the MapReduce architecture, the proposed scheduling method executes scheduling in parallel across distributed systems. Task prioritization in the map phase (using a fuzzy C-means clustering approach based on memory constraints) and optimum scheduling in the reduction phase (using an opposing grey wolf optimization algorithm) are the two primary operations that make up the MapReduce architecture. Making use of makespan, cost, and system usage, the performance of the suggested technique is examined.

In order to maximize resource efficiency, the authors of [25] suggest MOTS (a hierarchical multi-objective task scheduling system) cluster jobs using the K-means method in conjunction with a load balancing equation. Then, they use evolutionary algorithms to optimize the clusters in order to decrease makespan. The latter is accomplished by eliminating data transmission by delivering linked successive activities to a physical machine based on their status. Cloudsim has been used to simulate and test our approach. Our results demonstrate a 10% decrease in makespan and a 4% improvement in CPU efficiency when contrasted with Mai's RL method and Bugerya's parallel implementation technique. Also, as compared to Bugerya's approaches, the cost of transmitting information between sequential jobs is 10% lower. Our suggested task scheduling strategy is well-suited for use in distributed large data processing systems, both in terms of the outcomes and the fact that it is based on the iHadoop approach for parallel implementation.

2.1 Research gaps in existing literature

Network processing, pattern matching, and market analytics are just a few of the many areas that have benefited from big data clustering techniques, but there are still a number of areas that need more investigation and new approaches.

- Although Paknejad's Hadoop-based system has been useful in monitoring large-scale network traffic, optimizing scalability is still necessary, particularly with the mobile network landscape developing quickly. This is particularly true for scale network traffic analysis. Improving the system's capacity to handle and interpret data from network traffic effectively in the face of increasing data quantities might be the subject of future study.
- Using the Map-reduce architecture, Banerjee's FSM-Ho method provides a potential way to increase the efficiency of frequent subgraph mining. But further research into other methods is required to make the algorithm even more efficient and scalable, particularly when dealing with bigger data sets and more complicated network topologies.
- When it comes to processing large amounts of data, Qin's thorough examination of Map-reduce technology reveals its strengths and weaknesses, which is useful for big data clustering research. Handling various data kinds, scalability concerns, and other special challenges in big data clustering might be the focus of future study. Performance optimization and scaling of algorithms in distributed computing settings.
- Data clustering resilience and noise handling: Research by Li and Wen on multi-subspace representations and low-level subspace clustering models sheds light on how to make clustering methods more resilient. In cases where big data is accessible, data is diverse, and there are many dimensions, further study is required to provide methods that can reliably and accurately manage noisy data and enhance clustering outcomes.
- Creating automated cluster analyses for use in the cloud: There are still holes in the automation of cluster analysis in cloud computing networks, even if technologies for clustering huge data in parallel have advanced. Intelligent clustering algorithms that can adapt to new data patterns and network circumstances in real time might be the subject of future study aimed at enhancing performance and minimizing resource use. Enhance the quality of cloud-based application user experience. Better analysis of massive datasets across many areas will be possible if these knowledge gaps are filled up with current big data clustering technologies.

Table 1: Comparison analysis of the existing methods

Reference	Objectives	Findings	Limitations
Reference [19]	High-order PCM method for clustering huge dataset	Able to handle very big and diverse datasets safeguard sensitive data stored on cloud servers	• High memory usage • High response time
Reference [20]	Different clustering algorithms inside a cloud-based computing setting	Real-world datasets are supported	• Lack of security • Frequent data access in cloud
Reference [21]	Enhanced method (T.K-means) with the Hadoop platform's MapReduce computing architecture enables parallel computation	The optimal value for the cutting parameters is derived by mining the mapping connection between the parameters, roughness of the surface, and material removal rate	• Does not support security • Higher resource consumption
Reference [22]	Large-scale distributed power system's big dataset for Hadoop Cloud	Hadoop-based parallel processing approach for GIS data	• Leakage of security • Does not strong for large datasets
Reference [23]	K-means clustering approach is suggested to find advanced areas of initial foci and an initial number of bunches	In a fraction of the time and with much less expense, the suggested method puts on a very serious show.	• Less processing and clustering speed • Limited, not ideal for large datasets
Reference [24]	GWO for efficient memory constraint parallelized resource allocation and optimum scheduling	Task prioritization in the map phase and optimum scheduling in the reduction phase	• However, disadvantages include potential job losses and lack of security as roles are automated. Implementation costs and security issues around hackers accessing private data are also risks.
Reference [25]	MOTS (a hierarchical multi-objective task scheduling system) cluster jobs using the K-means method	Demonstrated a 10% decrease in makespan and a 4% improvement in CPU efficiency when contrasted with Mai's RL method and Bugerya's parallel implementation technique	• High latency • Low clustering strength

3 Methods and materials

Fig 1 shows the processes that make up a secure Hadoop cluster and the suggested design. Before being stored in HDFS, the input file is first encrypted and then split into N number of pieces. Once the encryption procedure is complete, the chunk files are placed in HDFS, which is comprised of a master node termed the name node and a slave node consisting of N number of data nodes. The encryption process employs the algorithm known as AES. When retrieving files, it is essential to utilize the K-means clustering map reduction to group similar chunks of data together.

3.1 Data chunking and encryption

NSL KDD 99 dataset– The problems with the current KDD Cup 99 dataset have been addressed in this newly suggested dataset for IDS experiments. For example, it ensures that neither the training set nor the test sets include duplicate entries. Each of the two sets of data—the training set contains 1,25,973 records and the test set 22,544 records—has 41 characteristics with values between 0 and 40. The number of occurrences of each attack type in the NSL KDD 99 dataset, as well as their distribution throughout the Train and Test files, are shown in Table I.

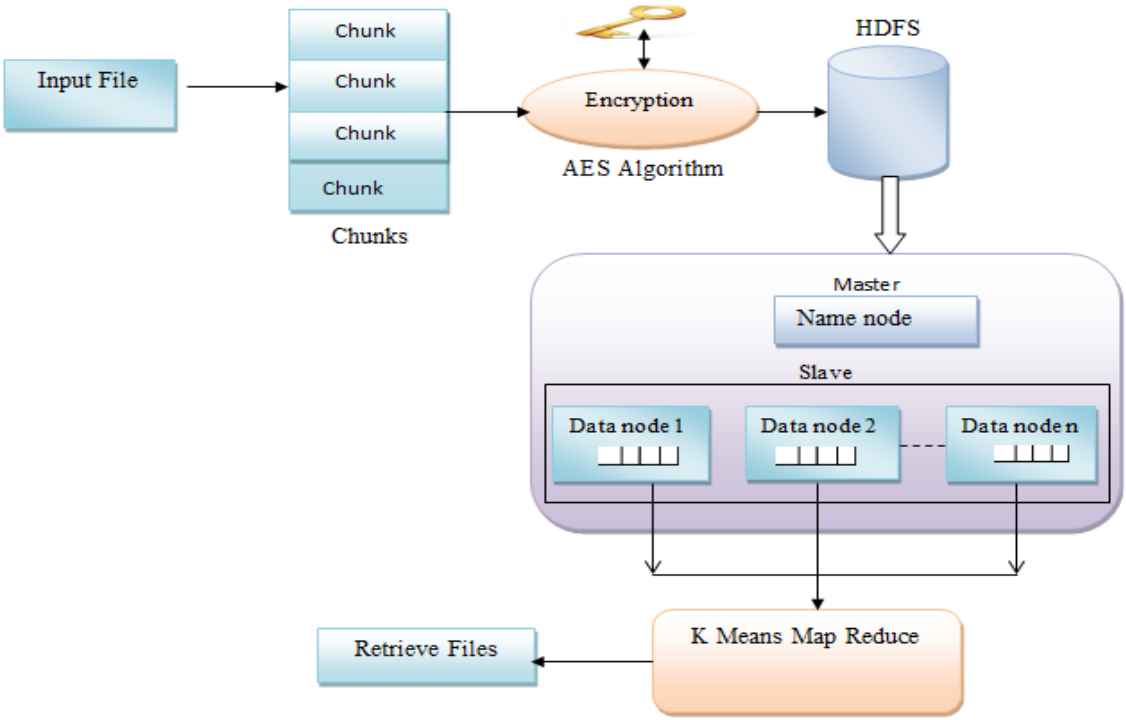

Figure 1: Proposed flowchart

Hadoop K-means clustering Map Reduce

Hadoop clusters are a subset of computer clusters that were developed for the purpose of storing and processing massive amounts of unstructured data in a decentralized setting. Commodity computers, which are inexpensive, power these clusters. Since the network connecting nodes in a Hadoop cluster is the only thing they have in common, this architecture is frequently called a "shared nothing" system. Racks are used to assemble large Hadoop clusters. It is preferable for network traffic to stay within a single rack rather than traverse many racks. One big problem with clustering methods is that they produce clusters regardless of whether the data really forms any. It is crucial to examine cluster tendencies before to processing and evaluate cluster quality after results. Nevertheless, the number of clusters (k) may be adjusted to alter cluster validation. We use the Silhouette clustering validation index to estimate the distance between clusters, and the Dunn clustering validation index to determine the distance inside clusters. The clustering procedure that is part of the suggested secure HDFS is shown in Figure 4.

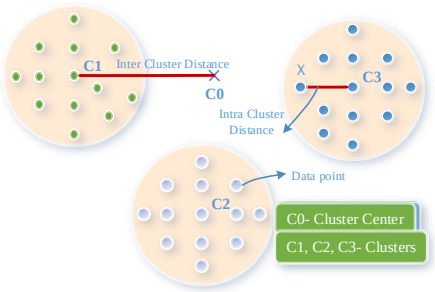

Figure 2: Hadoop clustering

The following is the formula for the inter-cluster distance, which is calculated by determining the mean distances between every cluster center and the global clustering center C 0.

$$Inter\_Cluster = \frac{1}{n}\sum_{k=1}^{m} D(C_0, C_k) \quad (8)$$

The ratio of the lowest and greatest distances from the cluster centers to each location is used to compute the intra-cluster distance.

$$Intra - Cluster = \frac{\min\limits_{k=1\ldots n} D(C_k, C_0)}{\max\limits_{k=1\ldots N} \max\limits_{X_i \in A_k} \{D(X_i, C_k)\}} \quad (9)$$

The Hadoop Map Reduce framework is a platform for handling massive amounts of data. Map reduction often partitions the starting data set into separate pieces, as seen in Fig. 2. It streamlines the procedure and does the map operation entirely in parallel. Each cluster node has a master and a slave that work together in the mapping process. Slave is in charge of the retrieval procedure for the master, while master has the indexing values. There is a job tracker on a single master server and task trackers on numerous slave servers. Below is a list of the primary components of map reduce:

- *Master node* – It takes user requests for jobs and controls the job tracker to make those requests a reality.
- *Slave node* – Wherever the user is at the moment, it can map, shuffle, and reduce.
- *Job tracker* – Since it keeps all of the users' scheduling knowledge, it is an entity that assigns tasks and monitors the jobs.
- *Task tracker* – The entity in question is responsible for keeping tabs on tasks and reporting work outcomes to the work Tracker.
- *Job* – Users have provided a comprehensive program.
- *Task* – The task at hand is work decomposition. As an example, a single user job may be divided into many jobs. A map-and-reduce function is applied to every job.
- *Task Attempt* – *The* specific effort at task execution in the slave node is known to it.

The functions involved in the process of map reduce are described below,

Map function:

Partitioning the data and running the Map operation utilizing distance calculation were the first steps in the map phase. Partitioned data is returned with key and value.

$$Map\ (k_1, v_1) \rightarrow (k', v')^{*} \quad (1)$$

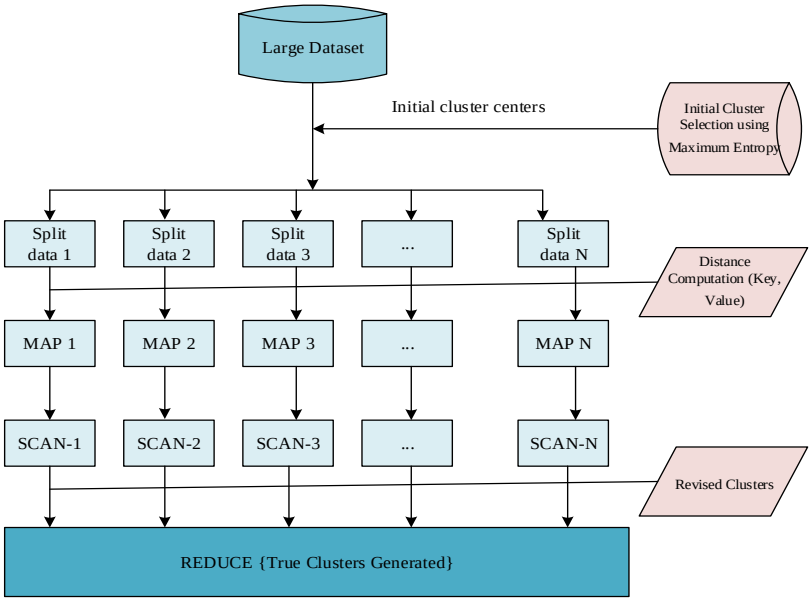

Figure 3: Map reduce function

Reduce function:
In this case, clusters of identical keys are created by minimizing the minimal and accurate (quality) reduction function. The following is an extension of the Reduce function class:

$$Reduce\ (k', (v'))^{*} \rightarrow (k', v')^{*} \quad (11)$$

The whole procedure might often be indicated as well, such

$$Map\ (k_1, v_1) \rightarrow List\ (k_2, v_2) \quad (12)$$

$$Reduce\ (k_2, list\ (v_2)) \rightarrow List\ (k_3, v_3) \quad (13)$$

In the master-slave paradigm of communication, one device exerts command over a chain of subordinate devices. In this model, one node plays the role of master while the other nodes play the role of slave. The idea behind this was to make sure that the data in the database was consistent. The master node gives the slave node instructions and gives them a task to do. The task will be processed and returned to the master by the slave nodes. A master node, or controller, in a real-time distributed system is primarily responsible for managing the tasks that are assigned to each slave device under its supervision. As soon as the master node runs out of juice, the slaves vote for a new leader. Figure 3 provides a comprehensive architectural diagram of the master/slave implementation.

4 Results discussion

Hadoop is the open-source software framework that our proposted system uses to store dispersed data. In this case, we shorten the time it takes to get data using the map-reduce paradigm by constructing an architecture with just two nodes: the node for names and the data node.

4.1 Simulation setup

We set up a single name node and a single data node in the Hadoop software environment to approximate the suggested work. Here we are deploying web apps on HDFS. To distribute the workload evenly between the two servers, we employ the Apache 2 load balancer. In our suggested solution, the workload is evenly distributed across the master and slave servers.

Table 2: Simulation configurations

Parameters		**values**	
Simulation Configuration	# Of Users	100	
	# Of Datacenters	10	
	# Of Brokers	1	
	Virtual Machines	Frequency	2 to 30 MIPS
		# Of VMs	10
		Speed of CPU	1000 MIPS
		RAM	512 MB
		Memory	100-1024 million bytes
		Bandwidth	100000
	Requests	# of requests	100 to 1000
		Bandwidth	1 to 3 mbps
		Memory	10 to 300 MB
		Request length	100 to 1600Ms
	HDFS	Memory per node in (GB)	32
		# Of worker nodes	≅ 5
		Data chunks	100mb per chunk
		master node	1
		Core per node	32
		Executor cores	1 per executor
		Executor memory (GB)	1 per executor
	Application server	Memory	10 MB
		Bandwidth	1000KBs
		MIPS	100
		RAM	10 to 15

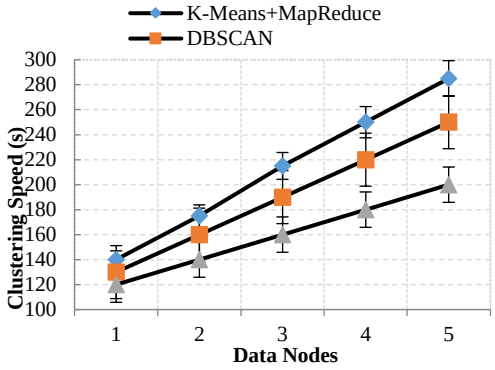

Figure 4: Clustering speed vs. data nodes

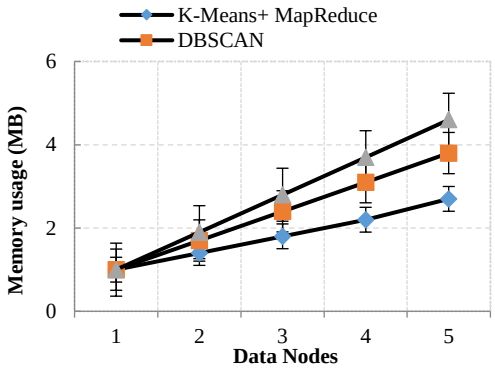

Figure 5: Memory usage vs. data nodes

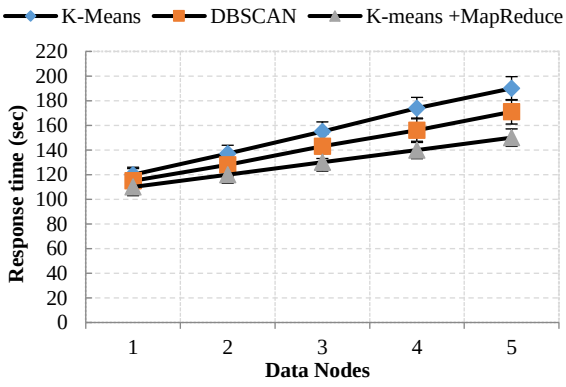

Figure 6: Response time vs. data nodes

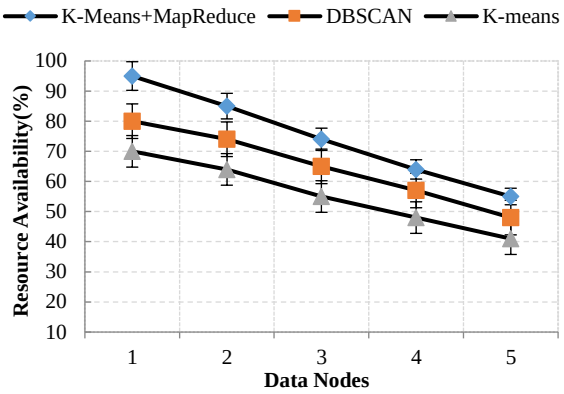

Figure 7: Resource availability vs. data nodes

Security breaches happen most frequently nowadays which can be found out by monitoring the server logs. This server-log analysis can be done by using Hadoop. This takes the analysis to the next level by improving security forensics which can be done as a low-cost platform.

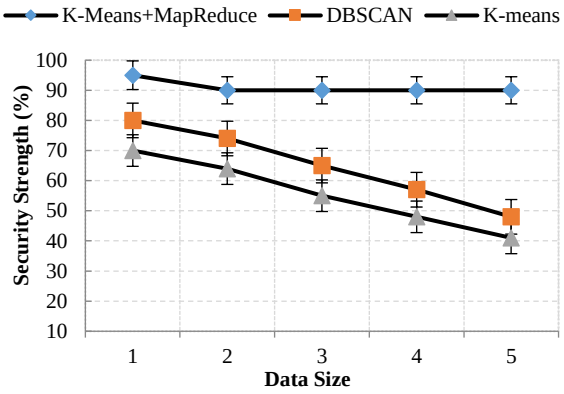

Figure 8: Security strength vs. data size

Hadoop MapReduce, the foundation of the massive data analysis approach, can handle enormous data sets in real time. Using student data on the prototype system, we verify the real-time performance. The AES-based key generation approach ensures system security by displaying excellent unpredictability and volatility. Also, the evaluation of time complexity is done for the proposed method. Table 3 illustrates the time complexity of the proposed method.

Table 2: Time complexity analysis

Process	Time Complexity	Description
AES	$O(N)$	Key generation bit length in bits
Encryption & Decryption	$O(\log n)$	n-Number of queries
Cluster Centre Selection	$O(1)$	Cluster threshold
Cluster formation & Storage	$O(n * Max * d * N)$	No of secure clusters stored in HDFS

5 Conclusion

Big Data is challenging to handle with standard technologies while also guaranteeing the security of the Big Data environment due to its huge size, variety of data kinds, and streaming nature. Big Data also has endless uses across several sectors. In this study, we bring forward a paradigm that addresses the computational complexity and security concerns associated with big data. Secure Hadoop clustering and quick data retrieval in a Hadoop cluster are two outcomes of our suggested approach. When storing chunks in HDFS, we use the AES algorithm to ensure their security. The AES algorithm encrypts every piece. The two nodes in this master-slave architecture are the data node and the name node, and they each have their own set of master and slave servers. We used K Means Map to show that our approach worked. Shortens the amount of time it takes to get data from HDFS.

5.1 Future work

There is always room for improvement in terms of the outcomes with regard to future enhancement. The K-Means method is able to provide more refined results since it selects the first set of centroids with fewer iterations than other algorithms.

Acknowledgement

This work was supported by Research on Digital Technology of Die Casting Dies for Knowledge Recognition and Reasoning(ygy2204) ; Construction and Key Technology Research of Intelligent Monitoring and Early Warning Platform for Urban Surface Water Environment Based on Stereoscopic Collaborative Perception System（G2022014116L）

References

[1] Andronie, M., Lăzăroiu, G., Karabolevski, O.L., Ștefănescu, R., Hurloiu, I., Dijmărescu, A., & Dijmărescu, I. (2022). Remote Big Data Management Tools, Sensing and Computing Technologies, and Visual Perception and Environment Mapping Algorithms in the Internet of Robotic Things. Electronics.

[2] Ullah, R., & Arslan, T. (2020). PySpark-Based Optimization of Microwave Image Reconstruction Algorithm for Head Imaging Big Data on High-Performance Computing and Google Cloud Platform. Applied Sciences.

[3] Li, Y., & Hei, X. (2022). Performance optimization of computing task scheduling based on the Hadoop big data platform. Neural Computing and Applications.

[4] Dandugala, L.S., & Vani, K.S. (2023). Big data clustering using fuzzy based energy efficient clustering and MobileNet V2. J. Intell. Fuzzy Syst., 46, 269-284.

[5] Salman, Z., & Alomary, A. (2023). Performance of the K-means and fuzzy C-means algorithms in big data analytics. International Journal of Information Technology, 1-6.

[6] Zhang, Y. (2021). The application of e-commerce recommendation system in smart cities based on big data and cloud computing. Comput. Sci. Inf. Syst., 18, 1359-1378.

[7] Jalalian, Z., & Sharifi, M. (2021). A Survey on Task Scheduling Algorithms in Cloud Computing for Fast Big Data Processing. International Journal of Information and Communication Technology Research.

[8] Zheng, Z. (2023). Phase space load balancing priority scheduling algorithm for cloud computing clusters. Automatika, 64, 1215 - 1224.

[9] Wang, J., & Zhao, B. (2021). Intelligent system for interactive online education based on cloud big data analytics. J. Intell. Fuzzy Syst., 40, 2839-2849.

[10] Sujit R Wakchaure, E.A. (2023). MR-AT: Map Reduce based Apriori Technique for Sequential Pattern Mining using Big Data in Hadoop. International Journal on Recent and Innovation Trends in Computing and Communication.

[11] Choudhary, A., Govil, M.C., Singh, G., Awasthi, L.K., & Pilli, E.S. (2022). Energy-aware scientific workflow scheduling in cloud environment. Cluster Computing, 25, 3845 - 3874.

[12] Bu, L., Zhang, H., Xing, H., & Wu, L. (2021). Research on parallel data processing of data mining platform in the background of cloud computing. Journal of Intelligent Systems, 30, 479 - 486.

[13] Ma, Y., & Wan, Y. (2021). 2021, Data Analysis Method of Intelligent Analysis Platform for Big Data of Film and Television. Complex., 9947832:1-9947832:10.

[14] Renugadevi, T., Geetha, K., Prabaharan, N., & Siano, P. (2020). Carbon-Efficient Virtual Machine Placement Based on Dynamic Voltage Frequency Scaling in Geo-Distributed Cloud Data Centers. Applied Sciences.

[15] Djafri, L. (2021). Dynamic Distributed and Parallel Machine Learning algorithms for big data mining processing. Data Technol. Appl., 56, 558-601.

[16] Wu, J.M., Li, R., Wu, M., & Lin, C. (2023). Mining skyline frequent-utility patterns from big data environment based on MapReduce framework. Intell. Data Anal., 27, 1359-1377.

[17] K. Kavitha, B. Sujatha, K.G. (2024). Analysis on Privacy Preserving Clustering Methods for Big Datasets Using Random Number Generators. Journal of Electrical Systems.

[18] Qiu, Z., Chen, R., & Yan, M. (2020). Monitoring Data Analysis Technology of Smart Grid Based on Cloud Computing. IOP Conference Series: Materials Science and Engineering, 750.

[19] Nanthini, V., & Mahalakshmi, R. (2024). PPHOPCM Privacy-Preserving High-order Possibilistic C-Means Algorithm for Big Data Clustering with Cloud Computing. International Journal of Advanced Research in Science, Communication and Technology.

[20] Gupta, V.K., Gupta, A., Agrawal, M., Sardana, A., Shukla, S.K., & Joshi, K. (2023). An Effectual Method for Clustering of Bigdata on the Cloud using Fuzzy c-means Algorithm. 2023 Global Conference on Information Technologies and Communications (GCITC), 1-5.

[21] Wei, X., Sun, Q., Liu, X., Yue, C.X., Liang, S.Y., & Wang, L. (2021). Research on parallel distributed clustering algorithm applied to cutting parameter optimization. The International Journal of Advanced Manufacturing Technology, 120, 7895 - 7904.

[22] Zhao, L., Wen, X., Huang, Z., Wang, N., & Zhang, Y. (2023). Power Big Data Analysis Platform Design Based on Hadoop. Journal of Physics: Conference Series, 2476.

[23] Kushwah, J., Jaloree, S., & R.S. Thakur (2020). Clustering of Multidimensional Big Data using Enhanced K-Mean Algorithm. International Journal of Innovative Technology and Exploring Engineering.

[24] Tukkoji, C., & Seetharam, K. (2020). Memory constraint parallelised resource allocation and optimal scheduling using oppositional GWO for handling big data in cloud environment. Int. J. Cloud Comput., 9, 432-452.

[25] Jalalian, Z., & Sharifi, M. (2021). A hierarchical multi-objective task scheduling approach for fast big data processing. The Journal of Supercomputing, 78, 2307 - 2336.