Available Online at www.ijcsmc.com/journal

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 11, Issue. 11, November 2022, pg.109 – 130

A Dynamics of Machine Learning on Map-Reduce Architecture for Enhancing Big Data Analysis Performance

[1]Ananda Khamaru; [2]Dr. Tryambak Hiwarkar
[1]Research Scholar; [2]Professor
[1,2]Department of Computer Science and Engineering
[1,2]School of Engineering and Technology
[1,2]Sardar Patel University Balaghat MP

DOI: https://doi.org/10.47760/ijcsmc.2022.v11i11.009

Abstract:

Big data is a fantastic resource for disseminating system-generated insights to external stakeholders. However, automation is required to manage such a large body of information, and this has spurred the development of data processing and machine learning tools. Just as in other fields of study and business, the ICT industry is serving and developing platforms and solutions to help professionals treat their knowledge and learn automatically. Large companies like Google and Microsoft, as well as the Apache Foundation's incubator, are the primary providers of these platforms.

Spark is an open-source platform for handling Big Data insights that have been tainted by contamination. This unified framework provides a variety of methods for dealing with unstructured or structured text data, graph data, and real-time streaming data. Spark relies on MLlib to create customised ML algorithms. To parallelize a huge cluster of machines for data

© 2022, IJCSMC All Rights Reserved

analytics, these methods require less memory, less processing time, and, to a large extent, hand tuned specialized architecture.

Data sets are analysed with machine learning methods including Linear Regression, Decision Tree, Random Forest, and Gradient Boosting Tree. In order to comprehend the data sets with the help of machine learning algorithms and to determine the best forecast value from the comparative study, the prediction model provided in this research is used.

One key goal of this study is to use the proposed model to make the most accurate forecast possible utilising machine learning methods. The suggested model utilizes the Apache Spark framework to perform a comparative analysis of the various existing approaches that have implemented the supervised and unsupervised techniques utilizing the MapReduce approach. By comparing the temporal complexity of each method, this method calculates the best prediction from the model. This dissertation emphasizes the characteristics of datasets that are most useful for examining the most effective prediction using machine learning algorithms.

I. Introduction:

Massive social implications have resulted from the use of machine learning (ML) systems in many fields and industries, including but not limited to computer vision, voice processing, natural language comprehension, neurology, healthcare, and the Internet of Things. In ML, the focus is on the question of "how to design a process architecture that improves progressively via experience" [1]. Learning from prior experience with regard to a set of tasks and an associated performance metric is what's known as an ML problem. Users may uncover previously unknown structure and make predictions from massive data sets using ML techniques. Advanced learning methods, large amounts of data, powerful computers, and a conducive environment all contribute to ML's success. Formation of machine learning is seen in Figure 1.

Information volume, retrieval speed, information kind, type, and organisation, information truthfulness, and information value are the five qualities that have been used to define big data (insights and effect). We arranged the five metrics into a hierarchical stack with ever more substantial data and value tiers. The lower layers (such as volume and speed) are more reliant on mechanical breakthroughs, whereas the higher layers (such as value) are better suited to applications that harness the vital energy of massive information. Existing ML standards and

© 2022, IJCSMC All Rights Reserved

computations should be updated to comprehend the estimate of big data analytics and to process such massive data properly.

What is Big Data?

As its name implies, "big data" refers to a substantial quantity of information. Big Data refers to a data set that is both extensive in size and intricate in structure. Big Data is unmanageable by conventional data processing tools due to its sheer size and complexity. Big Data refers to huge, varied datasets that can be either organised or unstructured. With the help of Big Data Analytics, businesses may tackle the difficulties they're encountering and find workable solutions. Businesses today attempt to wade through vast amounts of data in search of patterns and insights that may be used to address pressing issues.

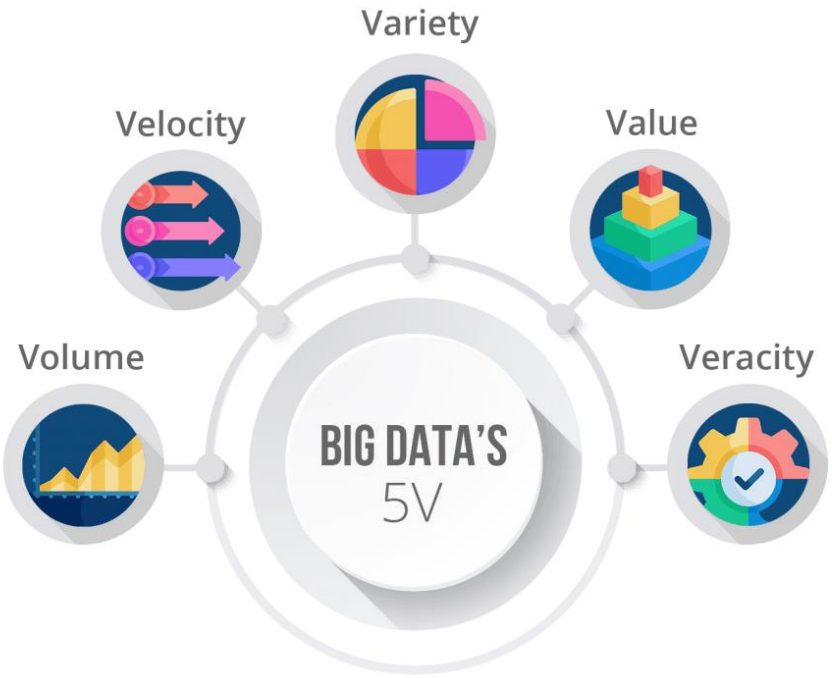
Although businesses have been amassing massive amounts of data for decades, the term "Big Data" didn't become widely used until the early to mid-2000s. Companies understood the significance of the massive amounts of data being collected every day.

What are the 5 Vs of Big Data?

As its name implies, "big data" refers to a substantial quantity of information. Big Data refers to a data set that is both extensive in size and intricate in structure. Big Data is unmanageable by conventional data processing tools due to its sheer size and complexity. Big Data refers to huge, varied datasets that can be either organised or unstructured. With the help of Big Data Analytics, businesses may tackle the difficulties they're encountering and find workable solutions. Businesses today attempt to wade through vast amounts of data in search of patterns and insights that may be used to address pressing issues.

Although businesses have been amassing massive amounts of data for decades, the term "Big Data" didn't become widely used until the early to mid-2000s. Companies understood the significance of the massive amounts of data being collected every day.

© 2022, IJCSMC All Rights Reserved



Types Of Big Data

Following are the types of Big Data:

1. **Structured**
2. **Unstructured**
3. **Semi-structured**

Structured:

"Structured" data refers to any information that has a predetermined structure that allows for its storage, retrieval, and processing. Talented computer scientists have made considerable strides over time in devising methods for working with this type of data (when the format is fully understood in advance) and extracting value from it. Unfortunately, we may now anticipate problems when the size of such data expands to a significant level, with typical quantities being in the range of many zettabytes.

Unstructured

Unstructured data refers to information that lacks a defined format or organisation. Besides the obvious difficulty in managing such a massive quantity of data, processing unstructured data presents a number of obstacles that prevent its value from being extracted. Unstructured data

© 2022, IJCSMC All Rights Reserved

often consists of a variety of file types, including text documents, photos, videos, and other media. Even while modern businesses have access to vast amounts of data, much of it is in an unstructured manner and so is of little use.

Semi-structured

Each type of data can exist in semi-structured information. We can think of semi-structured data as structured, but it is not specified in the same way that a table is in a relational database management system. Data stored in an XML file is an example of semi-structured data.

1.1 HADOOP'S FEATURES FOR BIG DATA ANALYTICS

Every business nowadays has numerous difficulties, most notably in evaluating the constant flow of data. However, data analytics takes the most time when reading and writing massive amounts of data using numerous discs in parallel mode. Big data provides an efficient solution to these problems. 7

Hadoop, in a nutshell, is a dependable data analysis and storage solution. Hadoop, a distributed file system, provides efficient storage, while the MapReduce paradigm handles data analysis. High-Performance Computing (HPC) systems are utilised to deal with the huge data. Message Passing Interface is a generic term for the APIs used by HPC systems to exchange data with one another.

HPC file systems are typically hosted on SANs so that all nodes in a cluster may easily access them. High-Performance Computing (HPC) file systems may handle complex computations. Network bandwidth issues arise as these technologies are used. When the data being transmitted is too vast, the employed bandwidth and the entire system collapse, causing performance concerns due to the computational nodes having to sit idle for an extended period of time.

One of the first issues to address is how to deal with hardware failure at individual nodes. As a part of its Hadoop Distributed File System (HDFS), the Hadoop project duplicates data three times and stores it in separate places. Replication is a common method used to protect data from being lost. With the redundant copies on hand, you won't lose any data if something goes wrong. Based on the RAID principle, this method ensures the HDFS is available at all times.

© 2022, IJCSMC All Rights Reserved

As for the second issue, it's the massive analytical job that necessitates combining data from different discs. The consolidation of dispersed and extracted data from various sources is currently the most difficult obstacle to overcome. A real solution to the issue of redundant disc reads and writes is the Spark framework, which is replacing the MapReduce method.

II. Literature Survey:

MapReduce plays an involuntarily action that parallelizes and executes the program on a large cluster of commodity hardware. Being a data processing tool, large amount of data is processed on several machines using distributed environment. Hadoop an open source distributed file system implement MapReduce for efficient storage on parallel databases. Though MapReduce programming model has mentionable benefits the following segment describes few limitations that affect its performance on clusters.

- Burhan U.I.Khan., et al [1] MapReduce is mainly used in Google for sorting, data mining and Machine Learning. This programming model deal with computational fault-tolerance in parallel distributed background. Locality optimization facilitated with read/write data operations to local disk saves network bandwidth. Redundant execution reduces the impact of slow machines, machine failures and data loss

- Erik R et al [2] A Comparative study on MapReduce with other database structures was discussed in this paper. Startup latency, Data shuffling, Textual formats, Natural indices and unmerged output are significant advantages. MapReduce provides a heterogeneous storage system for data loading with fine grain fault tolerance and execution of more complicated function through simple SQL queries.

- Douglas T et al [3] The MapReduce implementation relies on an in-house cluster management system for distribution and execution of user tasks on shared machines. This cluster management resembles the other systems such as Condor [3]. Redundant execution that recover data loss caused by failures and locality aware scheduling to reduce the amount of data sent across congested network links are fundamental similarities targeted in MapReduce programming model.

- Soila K et al [5] Locality optimizations are motivated from techniques such as active disks where the computation is channelized into processing elements which are adjacent to local disks. This technique condenses the amount of data sent across I/O subsystem or the network. This system support commodity processor in which the small number of disks is directly connected instead of running directly on disk controller processor.

- John B et al [6] This paper compared the performance of MapReduce and parallel databases. The comparative results addressed the shortcomings of MapReduce model that targets mainly on engineering considerations, reading unnecessary data, merging results and data loading.

© 2022, IJCSMC All Rights Reserved

• Huston L et al [7] MapReduce comprises of row-oriented fashion to scan the input data that apparently slow the execution of analytical tasks when compared to other advanced knowledge databases in interactive searches. In Trojan layout data is organized inside data blocks according to the incoming workloads implemented on top of HDFS. This layout replaces the MapReduce framework due to specific features such as co-locating attributes according to query workloads, different data block replicas and mimic fractured mirrors.

• Sijie G et al [9] Mastiff is optimized system for time based big data analytics. It exploits a systematic combination of a column group store and a light weight helper structure for diverse workloads to boast query performance. The MapReduce framework is replaced by this system for its high data loading scheme and high query performance.

• Herodotos H et al [10] Hadoop software stack consists of MapReduce execution engine, pluggable distributed storage engine and a range of procedural to declarative interfaces for Big Data analytics. Negative aspect of Hadoop performance is explicitly proved by Starfish introduced by Herodotos. These self-tuning system lifecycle analytics automatically that parallel runs knob for performance tuning to optimize resources, time and money [10]

Google designed a scalable distributed file system for large data intensive applications. The aspect of fault tolerance on execution of commodity hardware delivers high aggregate performance to a large number of clients. Thus MapReduce facilitate high aggregate throughput to parallel readers and writers in a DFS.

III. MAP REDUCE MODEL

The MapReduce framework is a data-flow paradigm used in data-centric applications. When compared to the conventional high-level database techniques, our straightforward explicit data flow programming model is a clear winner [64]. The MapReduce paradigm employs clusters or grids to parallelize massive data collections. The two main parts of a MapReduce programme are the Map() and Reduce() routines. These procedures are implemented in computer clusters to process massive datasets.

The functions Map( ) and Reduce( ) are given below where K1 and K2 are two key pairs and V1 and V2 are value pairs.

$$\text{Map } (K_1, V_1) \rightarrow \text{list } (K_2, V_2) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad 3.1$$

$$\text{Reduce } (K_2, \text{list } (V_2)) \rightarrow \text{list } (V_2) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad 3.2$$

© 2022, IJCSMC All Rights Reserved

Managing the "infrastructure" is what a Map() method does to carry out filtering and sorting. Reducing() is a useful function for summarising the "framework"-building processes. We call this underlying structure a "infrastructure framework" since it supports other systems. This system manages data transfer between its numerous components in order to allow multiple jobs to operate in parallel on geographically dispersed computers. Information processing can be broken down into several distinct stages, as shown in Figure 3.1.

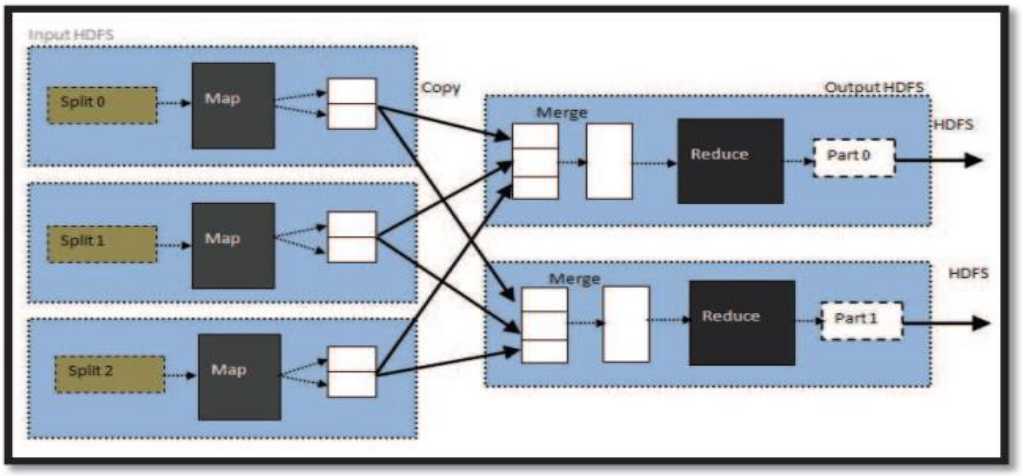


Fig: 1 Stages of Map Reduce

3.3 SPARK FRAMEWORK

Hadoop is widely used in business and the technological industries for analytics to meet the needs highlighted by certain datasets. MapReduce is a powerful tool for parallel programming that enables processing that is fault tolerant, flexible, scalable, and inexpensive.

When dealing with such systems, processing speed is of paramount importance. Large datasets are efficient for query processing and calculation, but they suffer from a lack of speed. Spark [65] overcomes the problems associated with the MapReduce programming approach, such as high disc rate, limited throughput, and declining cluster performance.

Apache introduced the Spark framework to facilitate the development of parallel distributed systems. It is not a component of the Hadoop ecosystem and operates independently. Spark's unique cluster management architecture improves storage performance [66]. A further benefit of Spark is its high-powered CPU, which allows for quick calculations.

© 2022, IJCSMC All Rights Reserved

- Batch application
- Iterative algorithms
- Interactive queries
- Streaming

3.3.1 FEATURES OF SPARK FRAMEWORK

Among SPARK's many benefits are those listed below:

1. Swiftness: It executes an application on a Hadoop Cluster at a rate that is one hundred times quicker in memory and ten times faster on discs.
2. To process interactive queries, it keeps 80 high-level operators and supports APIs built in Java, Scala, R, and Python.
3. Spark Framework supports advanced analytic methods like query analysis, machine learning, streaming analytics, and graph analytics.
4. There are three examples of Spark deployment, and they are as follows.
5. Spark as a Standalone: Spark is built upon the HDFS concept to handle all Spark tasks in parallel across a cluster.
6. Spark may be deployed on Yarn without the need for any prior installation of Hadoop, and it is also compatible with the rest of the Hadoop ecosystem.
7. Spark in MapReduce: Spark job launch begins immediately upon launch, without the need for any additional administrative privileges.

3.3.2 SPARK EXECUTION MODEL

When a Spark application is being executed, many runtime perceptions such as driver, executor, task, job, and stage are involved. Code written in Spark needs to be resource-efficient, so it's important to understand each part. Spark applications are mapped to a single driver and a collection of executors for processing over a cluster of servers.

At runtime, the driver is responsible for managing the application's implementation by allocating resources and assigning tasks. In a distributed system, jobs are started by the driver, but the shell takes over as the driver if the driver is not present. Cache memory maintenance and task execution are under the purview of the executors. Based on how much data is allocated in real

© 2022, IJCSMC All Rights Reserved

time, the lifetime of an executor can change. Only a limited number of additional execution slots are kept by this executor task.

IV. USING SPARK SQL

Spark SQL is a database query processing tool. SQL and the Spark DataFrame API are used to access it from within Spark applications. The SQLContext class, or a descendent thereof, serves as the entry point to all Spark SQL features. An SQLContext is used to convert an RDD, Hive table, or other data source into a DataFrame. Spark applications can access data stored in Hive by creating a HiveContext, a class that inherits from SQLContext.

By utilising a HiveContext, one may execute HiveQL queries and have access to tables stored in the Hive Metastore.

V. HADOOP DISTRIBUTED FILE SYSTEM

With today's advanced technology comes a deluge of data that's difficult for a single computer to process. Because of this, it is crucial to store the partitioned data blocks among a number of computers [68]. The term "distributed file system" is used to describe a data storage system that is shared across multiple computers.

The data are dispersed over the network in multiple storage sites. In network programming, it is common to face many challenges while attempting to manage and direct the flow of data. In a distributed system, for example, the most difficult obstacle is the occasional failure of a single node. HDFS (Hadoop Distributed File System) is the file system used in the Hadoop environment [68]. Name Node and Data Node are essential parts of HDFS. The Hadoop Distributed File System employs a master/slave setup. In a Hadoop Cluster, data is dispersed among racks. The cluster also contains a job tracker and a task tracker, which are two administrative tools [69]. Hadoop's internal architecture is depicted in Figure 3.2.

© 2022, IJCSMC All Rights Reserved

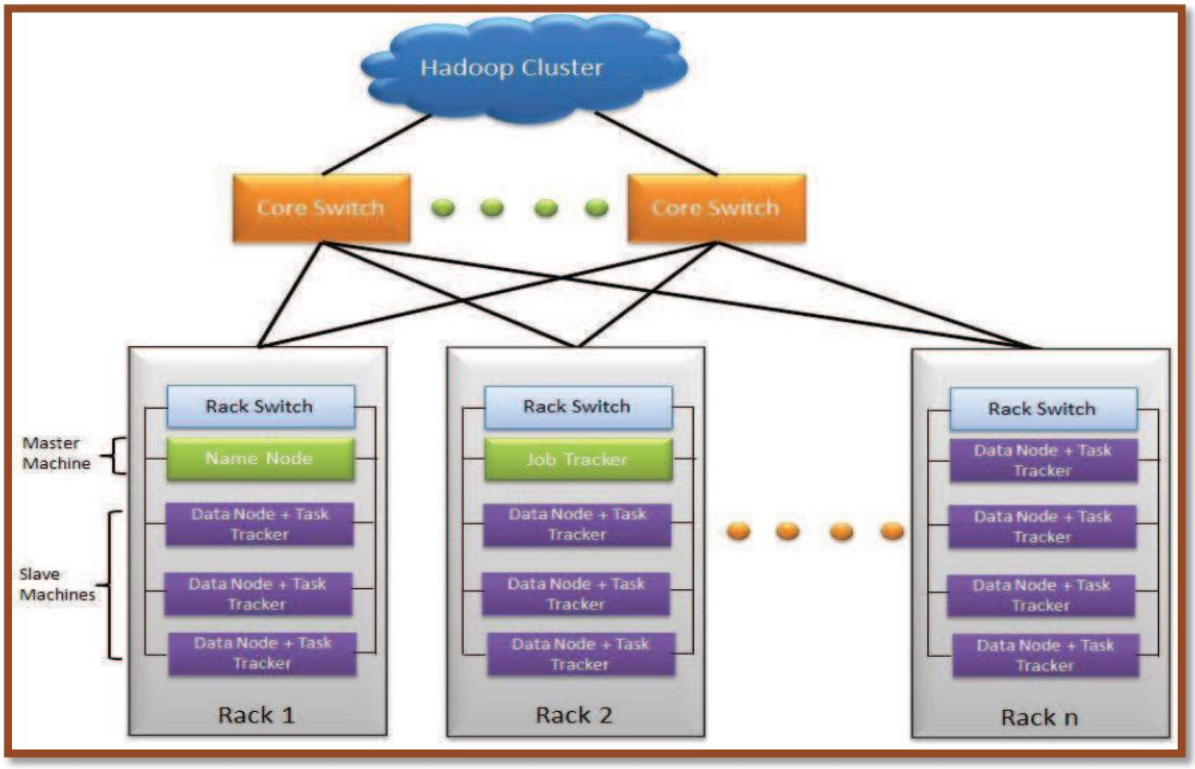


Fig:2 Hadoop Architecture

VI. MACHINE LEARNING TECHNIQUES

Machine learning is a branch of computer science that uses computers to learn from data [70]. Numerous potent algorithms for recognizing patterns, gaining understanding, and making predictions make up Machine Learning. The area of Artificial Intelligence (AI) emerged as a specialization dedicated to creating intelligent machines capable of acquiring new skills on their own. In this era of rapid progress, there is a famine of information due to the abundance of data.

There are two main ways to describe data: I in a formal schema, and (ii) informally.

Images, documents, text, music, and video are all examples of unstructured data, while tabular data is the opposite. By enhancing prediction and data-driven judgments, Machine Learning's analytic models aid in knowledge capture [71].

For tasks involving the analysis of streaming data, such as those performed by secure email clients, spam blockers, user-friendly text and speech recognition software, and effective web search engines, it is indispensable. Machine learning is a scientific application that generates reasoning by turning raw data into understanding [70]. To recognise patterns, gain

© 2022, IJCSMC All Rights Reserved

understanding, and make predictions, machine learning makes use of numerous potent algorithms. The field of Artificial Intelligence (AI) grew out of an earlier effort to create intelligent machines capable of acquiring new skills on their own. In this era of rapid progress, there is a famine of information due to the abundance of data. There are two main ways to represent data: I in a structured format, and (ii) in an unstructured format.

In contrast to the tabular data that makes up structured data, the unstructured data that makes up unstructured data consists of things like pictures, documents, text, audio, and video. By enhancing prediction and data-driven judgments, Machine Learning's analytic models aid in the knowledge capture process [71].

For tasks involving the analysis of streaming data, such as those performed by secure email clients, spam blockers, user-friendly text and speech recognition software, and effective web search engines, it is indispensable.

VII. ENVIRONMENTAL SETUP

Using Amazon EC2 instances, we provide a consistent environment across different cloud providers. The machine learning algorithms are executed in a Jupyter notebook, which was installed alongside Spark and Anaconda on the virtual computer. An open-source programme called "Orange" is used to decipher the cross-validation results of Machine Learning programmes.

The necessary implementation resources are shown in Table

Work Flow	Tools	Version
Job Distribution	Spark	Spark-2.1.0-bin-hadoop2.7
Prediction	Linear Regression Random Forest Decision Tree Gradient Boosting Tree	Anaconda3-4.3.1
Canvas	Orange	Orange 3.2

Table –1: Tools Applicable for Implementation

© 2022, IJCSMC All Rights Reserved

Figure 1 depicts the process that outlines the deployment of virtual machine instances on cloud platforms. The first stage involves accessing a data collection hosted in the cloud.

Putty, an open source key generator tool, is used to exchange RSA keys on a cloud system during the cloud environment's setup phase. An EC2 instance is made available after successful authentication. Tools helpful to the implementation task are installed on the instance as needed. With this infrastructure in place, the

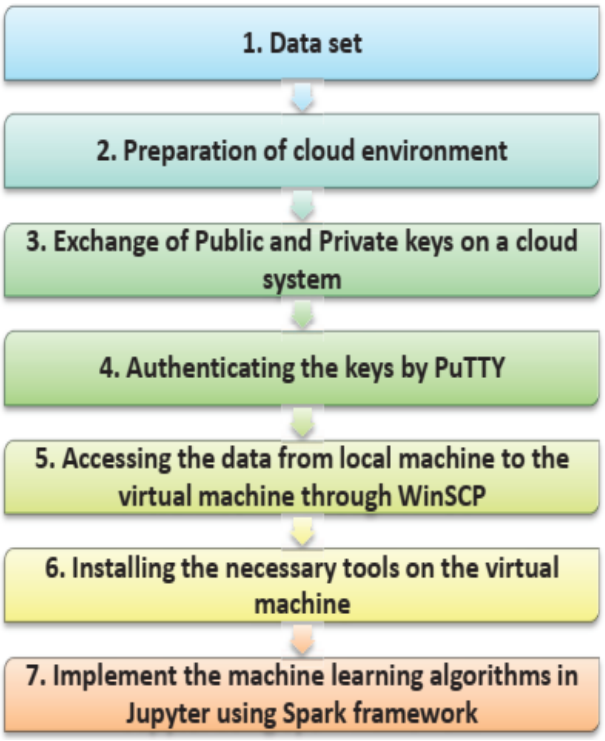


Figure – 3: Workflow of Environmental Setup on a Cloud Instance

VIII. TOOLS

By analysing test data, Machine Learning algorithms unwittingly record the traits they discover when examining datasets. Input data, features, and the expected result all make up what is called a training set. A continuous value or a forecast based on a class label is returned by the function.

The suggested work is implemented using Scikit Learn, a Python Machine Learning framework. You may use either supervised or unsupervised learning techniques with it. Another programme that may be used with the Apache Spark framework is the Jupyter notebook. This Spark framework makes use of Jupyter notebook to apply Machine Learning algorithms, and distributes workloads to virtual machines running the software.

© 2022, IJCSMC All Rights Reserved

Data loading and transformation are made easier with Orange, an open-source programme. Data mining, Machine Learning, predictive analytics, and statistical modelling all benefit from it. The programme analyses how several algorithms derived from the suggested model compare to one another.

IX. PHASES OF PROPOSED MODEL

In the suggested paradigm, there are three major steps. The first stage is dedicated to gathering data, which may include a summary or full description of the dataset. It then goes on to discuss the algorithm's description, workflow, and implementation in detail. In the third and final step, we compare and contrast the learning algorithms' outcomes and error visualizations.

- Phase I: Data Collection
- Phase II: Proposed Model and Algorithm
- Phase III: Comparative Result analysis

EXAMINATION OF THE SCHEME

Major regression models for prediction are deployed on two distinct frameworks, MapReduce and Spark, and their accuracy is analyzed using a common framework. Time and spatial complexity are calculated to assess overall performance.

DATASET ANALYSIS

The dataset that was used for this research project includes information that is connected to temperature. It consists of the average temperature for the entire year, computed from January to December each year. In addition to that, it takes into account the fluctuations brought on by the changing of the seasons throughout the course of the year. These are broken down into January to February, March to May, June to September, and October to December categories respectively. The dataset that was measured for analysis included temperature readings spanning one hundred and fifteen years, from 1904 all the way up to 2016. This information is put to use in the process of calculating and forecasting the years to come. The temperature dataset that is utilised as a training data set may be found

© 2022, IJCSMC All Rights Reserved

in Table 5.1. In order to provide accurate forecasts for future years, Machine Learning algorithms must first be trained on this data set. The average yearly temperatures are depicted graphically in Figure 5.1 for a number of different years.

Tab: Annual Temrature of Data

Year	ANNUAL
1904	19.51
1908	19.44
1912	19.25
1916	19.22
1920	19.03
1924	19.51
1928	19.08
1932	19.09
1936	19.13
1940	19.01
1944	19.31
1948	19.27
1952	19.09
1956	19.41
1960	19.64
1964	19.34
1968	19.02
1972	19.02
1976	19.37
1980	19.07
1984	19.54
1988	19.32
1992	19.36
1996	19.52
2000	19.24
2004	19.37
2008	19.3
2012	19.61
2016	19.4

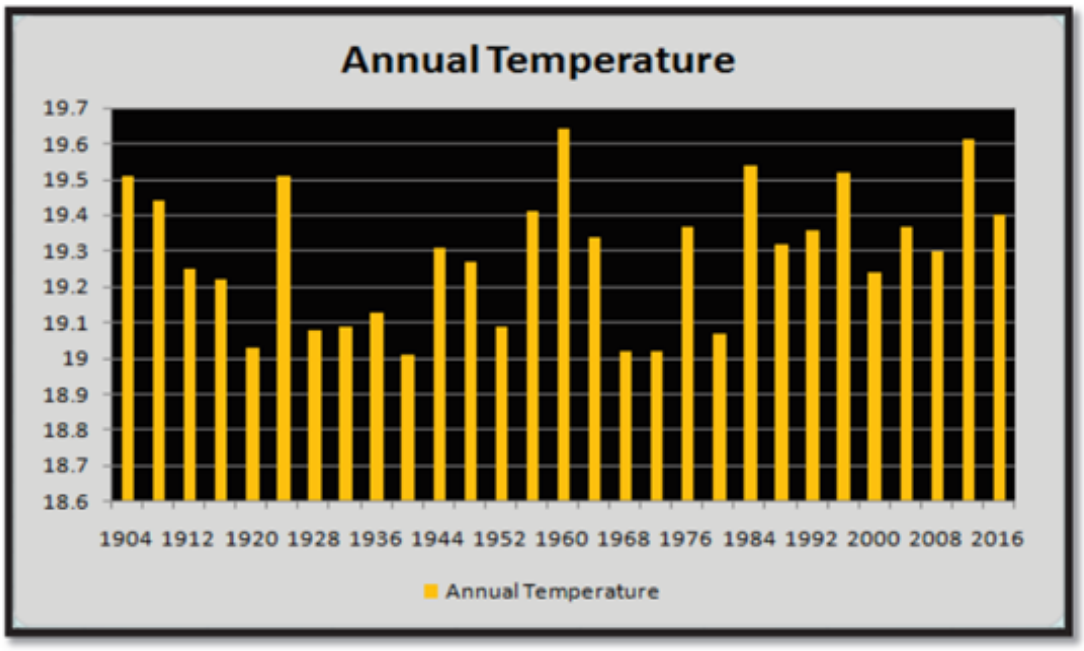


Figure – 4 : Year-wise Annual Temperature Data

© 2022, IJCSMC All Rights Reserved

X. MACHINE LEARNING ALGORITHMS FOR PREDICTION ON DATASETS

The study presented here includes a model that may be used to a number of different Machine Learning methods. The graphical representations used here are meant to demonstrate the predictions that were obtained by the learning methods. The following is a list of the numerous machine learning methods that may be used for prediction:

- Linear Regression
- Decision Tree
- Random Forest
- Gradient Boosting Tree

10.1.1 LINEAR REGRESSION ALGORITHM

The linear regression algorithm is used to implement the model. Figure 5.2 depicts a graphical depiction of data by comparing the yearly temperature with predicted.

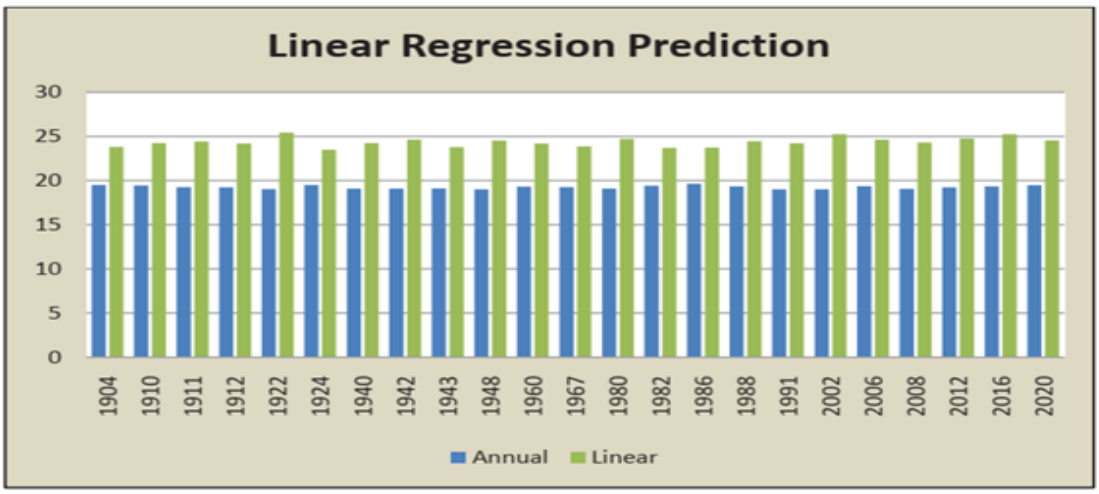


Figure – 5 Annual Temperature and Prediction Representation using Linear Regression Algorithm

Observation: Linear Regression computes predictions and RMSE. The prediction measures are less accurate and the graphical mapping shows wide deviations, which increases error rate with minimal time and space use. Section 5.5 discusses this algorithm's challenges.

© 2022, IJCSMC All Rights Reserved

Figure 5.3 shows yearly temperature predictions. It's the dataset's yearly temperature. Variance in predicted values increases mistake rate.

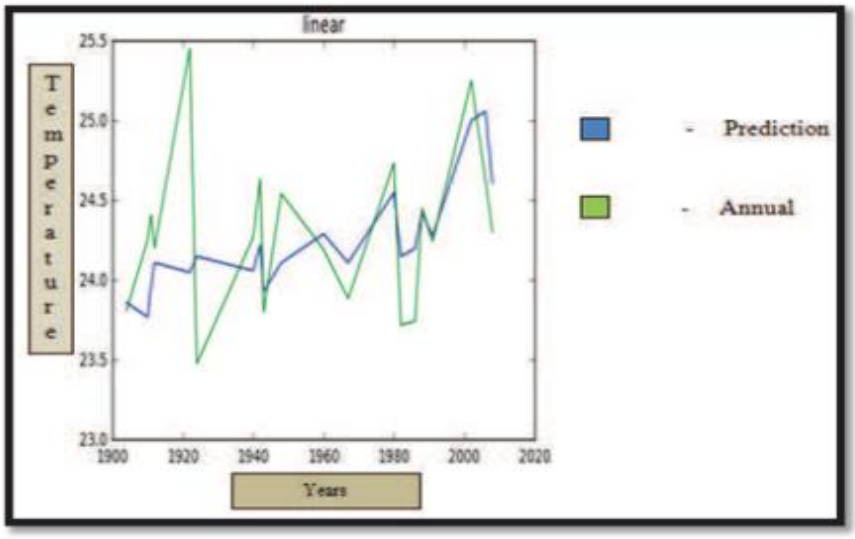


Figure – 6 : Annual and Prediction Value for Linear Regression Algorithm

10.1.2 DECISION TREE

Decision Tree algorithm predicts tree structure. .

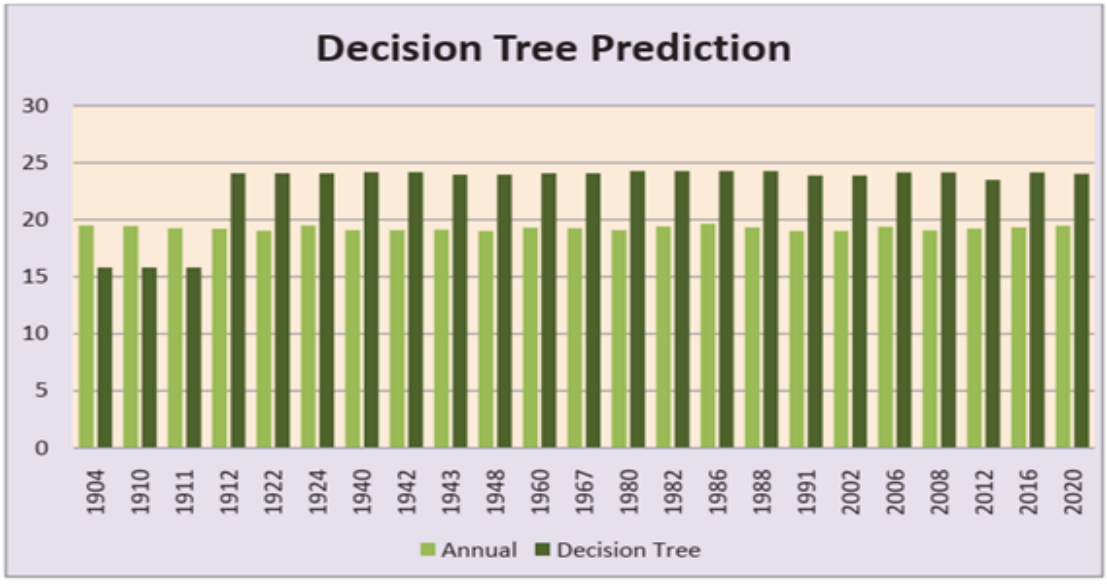


Figure – 7 : Annual Temperature and Prediction Representation using Decision Tree Algorithm

© 2022, IJCSMC All Rights Reserved

10.1.3 RANDOM FOREST

The tree structure was generated using the Random Forest technique, which was used to the prediction analysis. The differences may be seen in the graphical depiction that is provided below. The forest is constructed using a number of different decision trees. The random forest approach is used to determine how the characteristics compare to the average impurity. These contaminants have a negative impact on the computation output of each and every decision tree in the forest.

It is accomplished without making any data-related assumptions of any kind. Rankings are derived from a variety of features, which are then normalised according to the relevance they bring to the table. It does this by putting into effect a transformation mechanism that selects features according to a threshold that is defined after the model has been fitted.

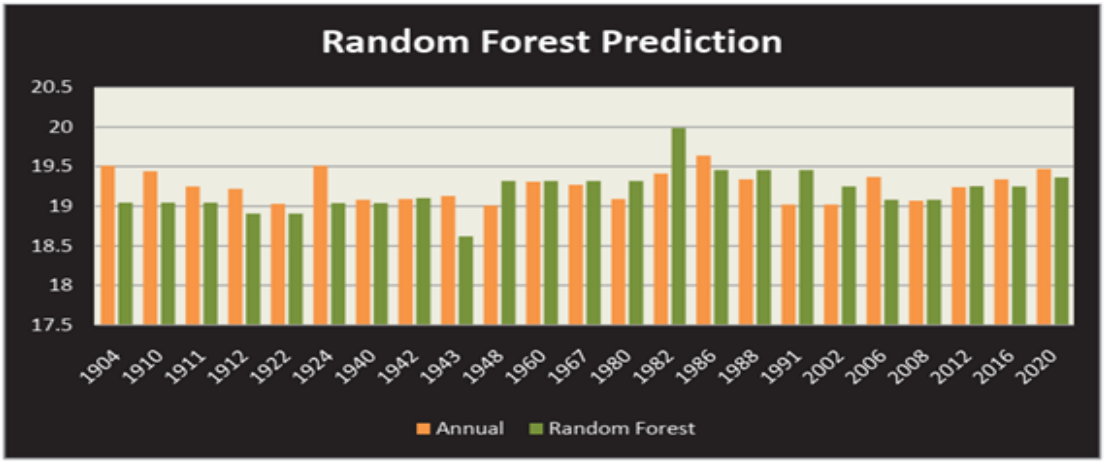


Figure – 8 : Annual Temperature and Prediction Representation using Random Forest Algorithm

Observations: The annual temperature is measured and compared to the prediction made by Random Forest, which has an accuracy of 95.33 percent. Figure 5.6 provides an illustration of a random forest that makes relevant predictions with regard to the yearly temperature. The results

© 2022, IJCSMC All Rights Reserved

of the random forest prediction method are acquired after generating a forest consisting of twenty decision trees. The ratio of space complexity to overall complexity is greater with the Random Forest approach.

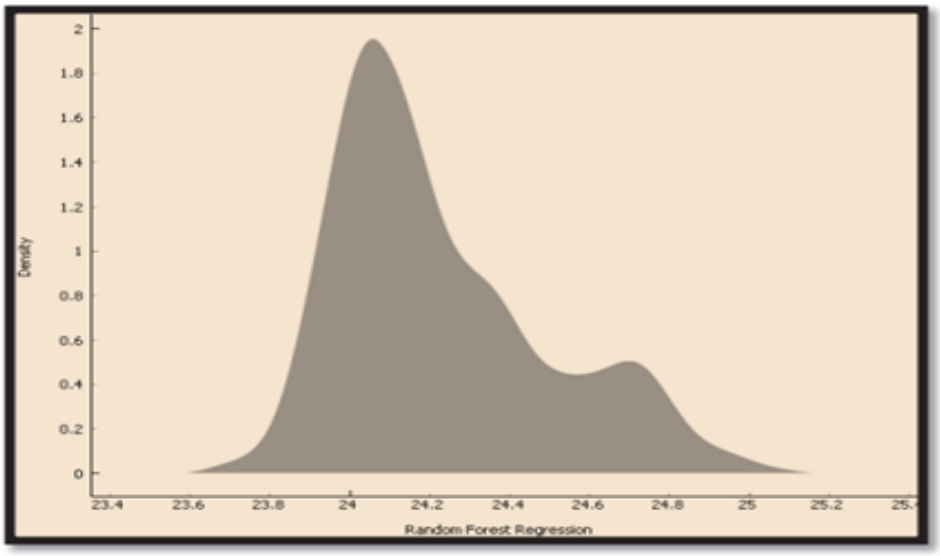


Figure – 9 : Random Forest Regression

The frequency distribution of the various forecast results is depicted in figure 5.7. The trees that make up a random forest are seen in Figure 5.8, along with the results of their predictions.

10.1.4 GRADIENT BOOSTING TREE

The decision tree is generated using a technique called Gradient Boosting Tree, and it is based on the prediction analysis. The differences may be seen in the graphical depiction that is provided below.

© 2022, IJCSMC All Rights Reserved

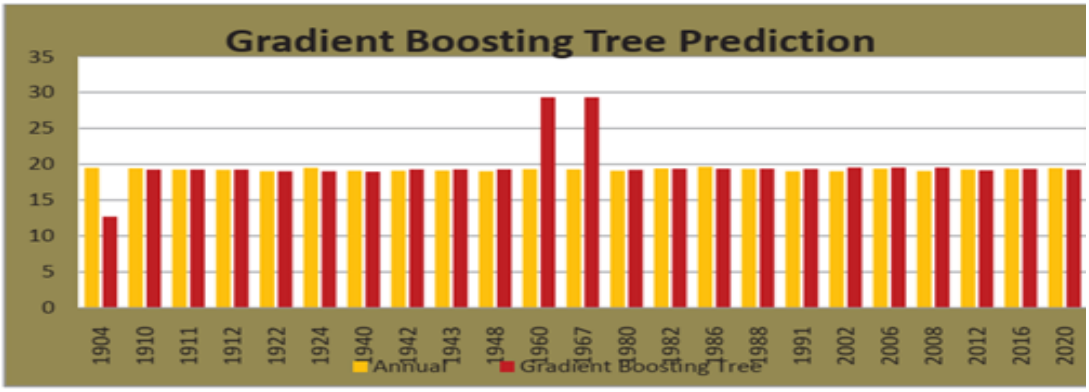


Figure – 10 : Annual Temperature and Prediction Representation using Gradient Boosting Tree

XI. COMPARATIVE ANALYSIS

In order to make accurate predictions, the prescribed model makes use of a number of machine learning algorithms. Some examples of these algorithms include linear regression, random forest, decision tree, and gradient boosting tree. The comparison findings of the various learning algorithms are

Year	Linear Regression	Random Forest	Gradient Booster	Decision Tree
1990	25.2511	19.2493	19.554	23.8818
1995	24.6349	19.08	19.554	24.1435
2000	24.3019	19.08	19.554	24.1435
2005	23.5327	19.457	19.475	24.517
2010	24.7825	19.251	19.147	23.486
2018	25.2378	19.249	19.357	24.143
2021	24.5673	19.364	19.264	24.0245

Observation: The results of comparing the yearly temperature to the forecast values of machine learning systems are displayed in Figure 5.11. The observed result demonstrates that the Random forest model accurately predicts the yearly temperature. When compared to other approaches to learning, it boasts an exceptionally high level of accuracy.

© 2022, IJCSMC All Rights Reserved

Conclusion:

Data analysis tasks that involve temporal and spatial elements might benefit from the systematic collection and evaluation of observations made using Machine Learning techniques. The given model uses the Spark framework to make the most of Machine Learning techniques in a decentralised setting. The goal is to minimise the computational and storage demands of future forecasts. Using the Spark framework on top of several Machine Learning techniques improves the model's efficiency by 70% compared to the MapReduce paradigm. An in-depth image of each learning algorithm, including a description of its individual features and an analysis of those features in light of the specified model, has been painted by the research. Several model dimensions are described here.

- ***Prediction:*** This developed model analyses the Indian temperature data set collected between the years 1901 to 2015. It predicts the temperature for the subsequent years from 2016 to 2020. This model is applicable for any prediction dataset.
- ***Time:*** This model evaluates the time of Machine Learning algorithms and traces out the utilized time by predicting the values on various algorithms.
- ***Space:*** This developed model is implemented on Machine Learning methods which require space as a factor. It also evaluates the space required for each algorithm.
- ***Spark Framework:*** This model can replace the existing MapReduce model of Hadoop system with Spark framework for reducing complexities.
- ***Job Allocation:*** This model distributes the job in a cluster and computes the time, space utilization and analytical performance.

The architectural foundation of this approach is the Spark framework, which plays a pivotal role in this context. PySpark's application to this paradigm aids in the simplification of the development of massively scalable machine learning algorithms. Machine learning algorithms are developed and analysed with improved execution plans over older techniques.

The advantages of optimization strategies like blocking and local aggregation, and the scalability of many different Machine Learning algorithms, are demonstrated empirically. With the advent of supplementary constructs to back up machine learning meta-tasks like model selection, researchers have been able to probe a broader set of approaches. The resulting data size is unprecedented.

© 2022, IJCSMC All Rights Reserved

References

[1] Burhan U.I.Khan., Rashidah F., and Hunain A. et.al, (2014) 'Critical Insight for MapReduce Optimization in Hadoop' in International J of Computer Science and Control Engineering, Vol.2, Issue-1, pp: 1-7.

[2] Erik R., Christos F. and David N. (2001) 'Active disk for large scale data processing', IEEE. J., pp 68 - 74 June.

[3] Douglas T., Todd T., and Miron L. (2004) 'Distributed computing in practice: The condor experience' in Concurrency and Computation Conference.

[4] Soila K., Jiaqi T., Rajeev G. and Priya N. (2009) 'An Analysis of Traces from a Production MapReduce Cluster' CMU-PDL-09-107 December.

[5] Soila K., Jiaqi T., Rajeev G. and Priya N. (2009) 'An Analysis of Traces from a Production MapReduce Cluster' CMU-PDL-09-107 December.

[6] Herodotos H., Harold L. and Liang D. et al. (2011) 'Starfish: a self-tuning system for Big Data Analytics' in Biennial Conference on Innovative Data System Research.

[7] Dean J. and Sanjay G. (2010) 'MapReduce: A flexible data processing tool' ACM communications Volume 53 pp 72 -77.

[8] Thomas H., Martin P.-Z. and Alexander S. (2012), 'Giant Data: MapReduce and Hadoop' University of Freiburg computer science department - Research Article.

[9] Burhan U.I.Khan., Rashidah F., and Hunain A. et.al, (2014) 'Critical Insight for MapReduce Optimization in Hadoop' in International J of Computer Science and Control Engineering, Vol.2, Issue-1, pp: 1-7.

[10] Tripti M., Neha Mangle, (2016), 'A Survey Paper on Big Data Analytics using MapReduce & Hive on Hadoop Framework' Special Issue on International Journal of Recent Advances in Engineering & Technology (IJRAET) Vol.4 I, Issue-2, pp: 112-118.

© 2022, IJCSMC All Rights Reserved