


RESEARCH

Open Access



# A canonical model for seasonal climate prediction using Big Data

M. P. Ramos<sup>1,2\*</sup> , P. M. Tasinaffo<sup>1</sup>, A. M. Cunha<sup>1</sup>, D. A. Silva<sup>1</sup>, G. S. Gonçalves<sup>1</sup> and L. A. V. Dias<sup>1</sup>

\*Correspondence:  
marcelopaivaramos@gmail.com

<sup>1</sup> Computer and Electronic Engineering Graduate Program, Brazilian Aeronautics Institute of Technology, São José dos Campos, São Paulo, Brazil  
Full list of author information is available at the end of the article

## Abstract

This article addresses the elaboration of a canonical model, involving methods, techniques, metrics, tools, and Big Data, applied to the knowledge of seasonal climate prediction, aiming at greater dynamics, speed, conciseness, and scalability. The proposed model was hosted in an environment capable of integrating different types of meteorological data and centralizing data stores. The seasonal climate prediction method called M-PRECLIS was designed and developed for practical application. The usability and efficiency of the proposed model was tested through a case study that made use of operational data generated by an atmospheric numerical model of the climate area found in the supercomputing environment of the Center for Weather Forecasting and Climate Studies linked to the Brazilian Institute for Space Research. The seasonal climate prediction uses ensemble members method to work and the main Big Data technologies used for data processing were: Python language, Apache Hadoop, Apache Hive, and the Optimized Row Columnar (ORC) file format. The main contributions of this research are the canonical model, its modules and internal components, the proposed method M-PRECLIS, and its use in a case study. After applying the model to a practical and real experiment, it was possible to analyze the results obtained and verify: the consistency of the model by the output images, the code complexity, the performance, and also to perform the comparison with related works. Thus, it was found that the proposed canonical model, based on the best practices of Big Data, is a viable alternative that can guide new paths to be followed.

**Keywords:** Big Data, Hadoop, Hive, MapReduce, Seasonal climate prediction, Atmospheric numerical model

## Introduction

The modern meteorology uses atmospheric numerical models to simulate the behavior and evolution of the atmosphere. The Center for Weather Forecasting and Climate Studies linked to the Brazilian Institute for Space Research (in Portuguese: *Centro de Previsão de Tempo e Estudos Climáticos—CPTEC do Instituto Nacional de Pesquisas Espaciais—INPE*) makes use of the supercomputer Cray XE-6 to run atmospheric numerical models operationally and in research, to predict weather, climate, and environment conditions.

Initially, it is possible to list several items that justified this investigative effort such as: the increase of spatial models resolution; the number of variables; the need to associate several models to obtain a probabilistic forecast with a greater number of combinations;

the use of observed historical data; and the combination of several types of data such as: radar, satellite, model, observation, among others. These were the main items that motivated this research involving Big Data.

Faced with a Big Data environment, it is necessary to consider some common challenges arising from expressive databases, taking into account some related issues like: the handling of large volumes and diversity of data; the data storage and retrieval; and the way of: how to analyze the entire volume of data involved, how to generate information in a timely manner, and also how to identify valuable information.

Ylijoki and Porras [1] conducted a research on the definitions of the term “Big Data”, in which 62 articles containing this term were listed between 1997 and 2016. The highest occurrences of definitions were: volume (59 occurrences in 62 analyzed articles), variety (55), velocity (46), value (17), and veracity (14). Figure 1 illustrates the timeline of Big Data definitions [1].

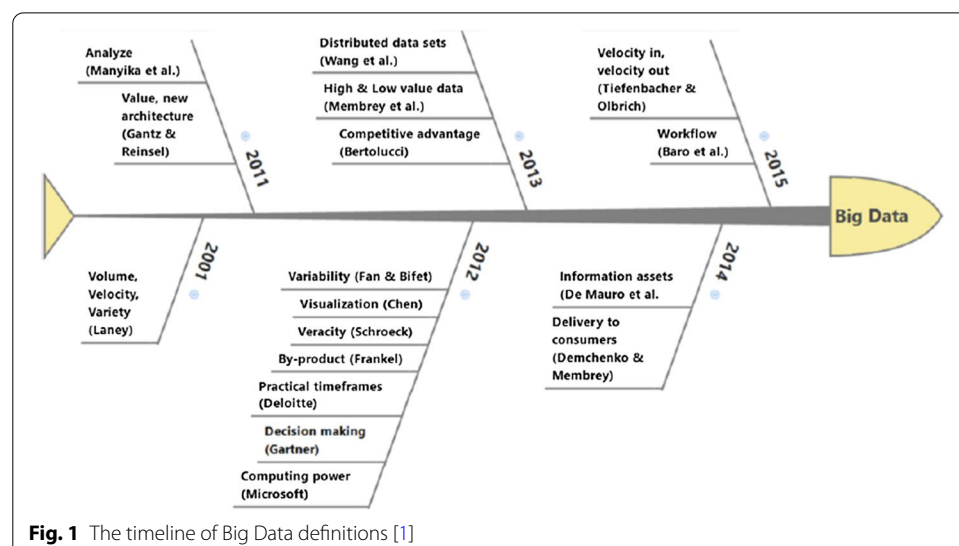
Laney [2] defined the three dimensions of “Big Data”: volume, variety, and velocity. Gantz e Reinsel [3] conceptualized the “value” characteristic and Schroeck et al. [4] established the “veracity” dimension. Therefore, the most used Big Data definition today by the scientific community is that of 5 factors/dimensions or 5 Vs: Volume, Variety, Velocity, Value, and Veracity [2–4], as shown in Fig. 2.

Big Data technologies permit the use of new methods, techniques, and tools to extract valuable information in a timely manner, allowing to process massive databases, combining considerable varieties of data. In this way, it becomes possible to increase the efficiency of processes and reduce the waste of resources involved, thus achieving greater speed to capture, discover, and analyze, if compared to the traditional model [5].

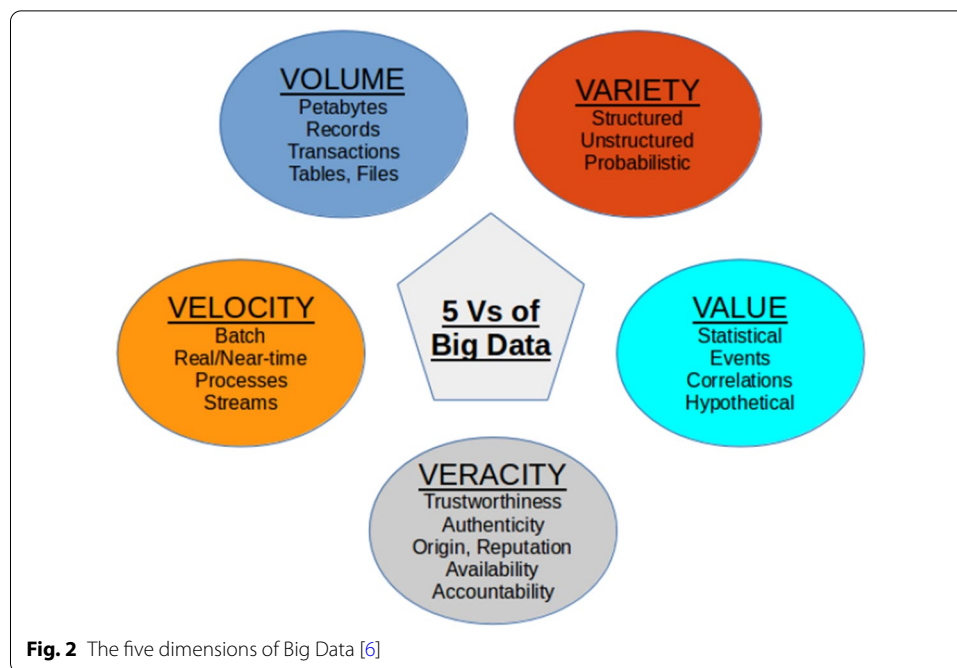
## Background

### The Big Data challenge

With the advent of the Internet of Things (IoT), there was an increase in access to Information Technology (IT) services and agile development of tools and applications, such as rapid technological and market changes, among others. At the same



**Fig. 1** The timeline of Big Data definitions [1]



time, the need arose to obtain more dynamic and flexible solutions for computing environments, which aim for quality, reliability, safety, security, economy, intelligence, and management simplicity.

Some barriers encountered have been the difficulty to monitor and manage: various computerized segments, storage systems, network equipments and services, as well as the difficulty to handle various tools configured in a decentralized and disintegrated way.

### The seasonal climate prediction

Weather is an ephemeral state of the atmosphere and the climate comprises a more comprehensive and stable period. Therefore, through the average study of weather on certain variables and locations, the difference between weather and climate is found on a time scale [7].

Although it is not yet possible to produce accurate weather forecasts beyond 1 week, it is already possible to predict probable future conditions based on averaging over the long term, usually a period between 1 month and 1 year. This probabilistic process is known as seasonal forecasting, generated by sets of predictions of climate models [8].

In an atmospheric numerical model, the atmosphere is projected on a grid-like plane and contains several vertical levels. Its mathematical equations allow to analyze the value of each point of the grid and also the interactions between layers, lateral quadrants, and also with the surface [9–13].

On the first stage of model execution, the use of meteorological data assimilation techniques are extremely important for the correction of inaccuracies in data that make up the initial and boundary conditions of the weather and climate forecast models. The work of Huang et al. is an example of applied research in this area [14].

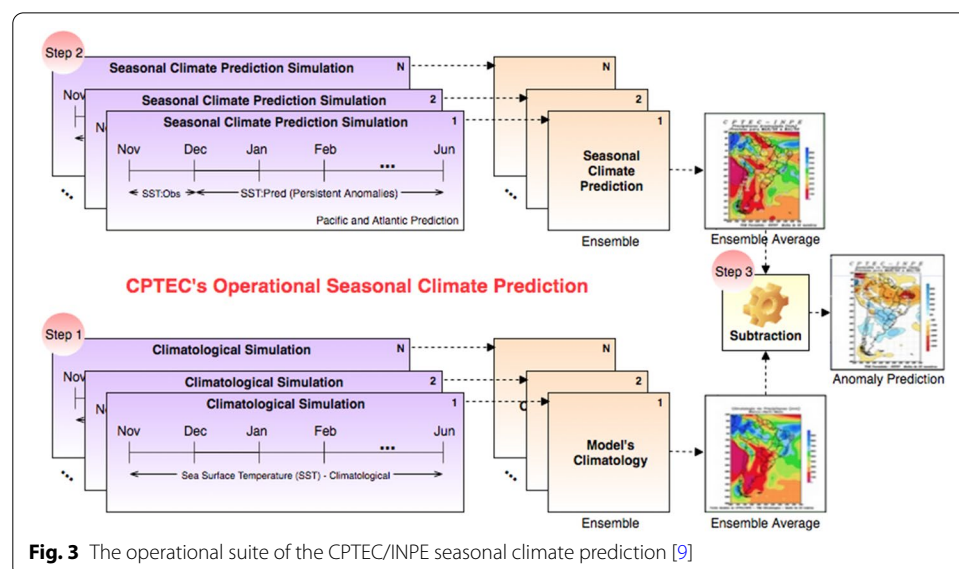
During this initial stage, data observed from equipments are interpolated with variables from previous forecasts and named “first guess”. In this first stage, there is also the computation of boundary conditions, sea surface temperature, sea ice, soil moisture, snow cap, greenhouse gases, and aerosol concentration, for example [9–13].

One method used to sample uncertainties associated with forecasts is known as a forecast by Ensemble members (sets). This method consists of a set of executions of a given model with small variations in the initial conditions of each execution and allows determining probabilities in certain scenarios. According to Lorenz, this methodology is based on the variability in which mathematical equations can result in small atmospheric variations, assuming that the model is perfect and only the analysis is disturbed [15–17].

A set of executions of an atmospheric numerical model is used in the process of seasonal climate prediction and forecasting of climatic anomalies. The first stage of this process consists of obtaining model’s climatology, that is, the monthly average of models forecasting for the processed period, where the model is executed over a long period of data of 30 years or more.

Then, the model is run on current input data to obtain the seasonal climate prediction for a future monthly or even annual period. Finally, it becomes feasible to conceive the forecast of climatic anomalies through the difference between the model climatology and the seasonal climate prediction, which results in a probabilistic forecast of the variables and locations with chances of suffering some variations of climate compared to a historic period [9].

Figure 3 illustrates the climate forecasting process using the Ensemble members method, where the atmospheric numerical model is executed “n” times with the necessary disturbances in its analysis file to obtain the climatology of the model and the seasonal climate prediction, also presenting the prediction of anomalies for a given period and location.



**Fig. 3** The operational suite of the CPTEC/INPE seasonal climate prediction [9]

### The Apache Hadoop

The Apache Hadoop project develops open-source software for reliable, scalable, and distributed computing. It is a framework that allows distributed processing and storage of data distributed across a cluster of computers. Its architecture is based on Big Data concepts, where a main node controls the other nodes, which offer processing and local storage [18, 19].

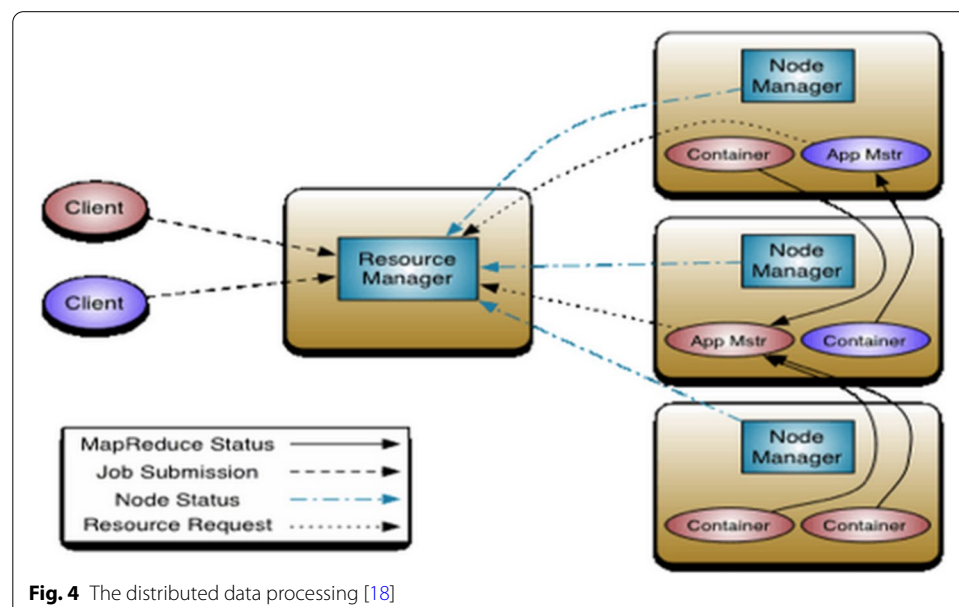
The distributed data processing is performed by the Hadoop Yet Another Resource Negotiator (YARN), responsible for resource management and scheduling tasks in the Cluster. Its main components are: ResourceManager, NodeManagers, ApplicationMaster, Containers, and JobTracker [18, 19].

The ResourceManager runs on the main node and is responsible for resource management and application scheduling. The NodeManager component is processed on all nodes and is responsible for the resources of the nodes and containers, for example, processors, memory, disk space, network, log, and others [18, 19].

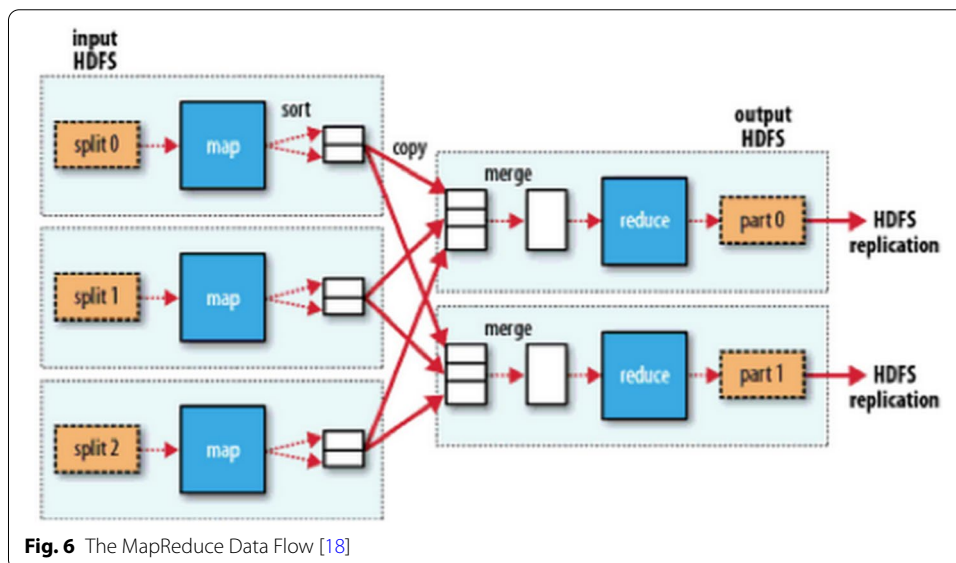
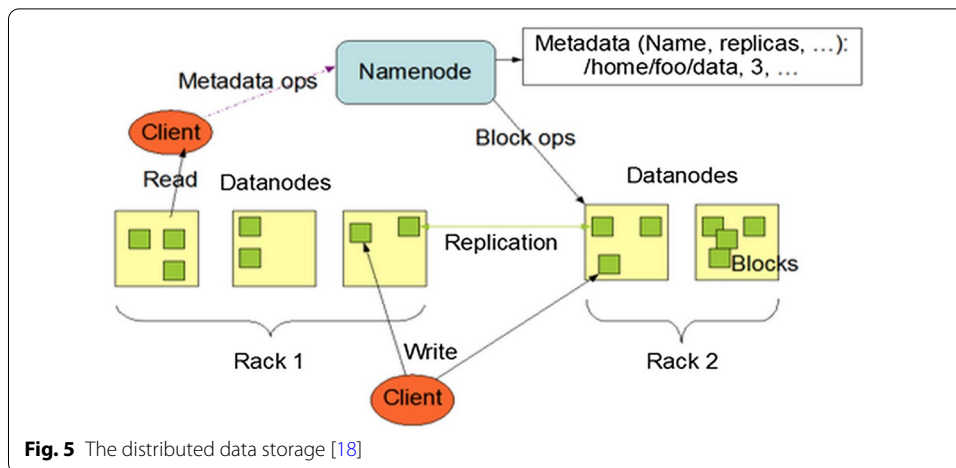
The ApplicationMaster is started only on slave nodes, where it will have one instance per application responsible for managing the number of Containers requested and processing the MapReduce application. Finally, JobTracker has a single instance on the main node and controls the progress of all applications [18, 19].

Figure 4 illustrates the architecture used by Hadoop to perform distributed data processing, in which the client requests resources to run an application to the Resource-Manager, which in its turn requests NodeManagers a sufficient number of nodes to create instances of ApplicationMaster and Containers, that communicate with the ResourceManager after starting [18, 19].

The Hadoop Distributed File System (HDFS) is the Hadoop distributed data storage system and its operation is based on a client/server architecture. The HDFS has three components: NameNode and SecondaryNameNode, that make up the server; and the DataNode present on all nodes in the cluster [18, 19].



**Fig. 4** The distributed data processing [18]



The NameNode is responsible for storing the metadata, relating the blocks of a file to the referring DataNode. The SecondaryNameNode has the function of monitoring and managing the NameNode, ensuring its availability. The DataNode represents the place where the blocks of a file are actually recorded [18, 19].

The Hadoop, when writing a file to HDFS, partitions it into one or more blocks and distributes them among DataNodes, as shown in Fig. 5. These blocks can be replicated in different DataNodes to provide fault tolerance [18, 19].

Figure 6 shows the MapReduce data flow, a model proposed by Google, which addresses a new programming paradigm for working with Big Data. This model allows the manipulation of Big Data in parallel and distributed way, in addition to providing fault tolerance, scaling Input/Output (I/O), and monitoring [18–23].

Map and Reduce are the two main operations of the MapReduce process, where at least six steps can be highlighted: (1) Input—when the text is stored in blocks in HDFS; (2) Splitting—when each block is divided into smaller parts, for example, text broken



down into lines; (3) Mapping—when each part (line) is computed for the key/value format; (4) Sort/Shuffle—when the Sort and Shuffle operations perform the sorting and grouping of data, according to the “key”; (5) Reducing—when calculating the values contained in each grouping; and, finally, (6) Output—when the result is recorded on the HDFS. Figure 7 shows the MapReduce flow applied to count words in a text [18–23].

### The Apache Hive

The Apache Hadoop has several benefits and technological innovations and brings some difficulties regarding its use. One of them is the fact that the developers are not familiar with the Map and Reduce technique, which requires considerable time to understand, and there are not many qualified professionals in the job market.

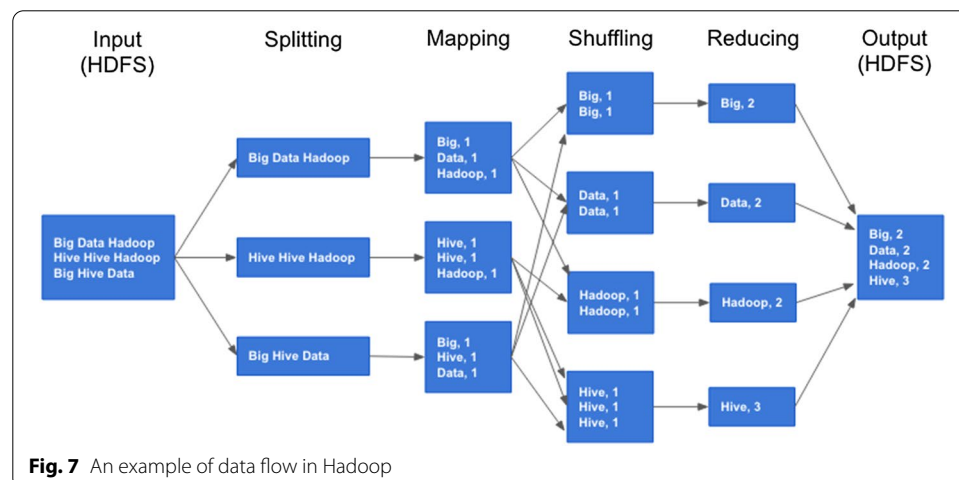
Another point is the complexity to perform operations and extract information from files stored in the HDFS. Even in simple queries, it is necessary to execute a considerable sequence of instructions. Then, the Apache Hive is used, in order to abstract the difficulties in using the Apache Hadoop [18–21, 24, 25].

The Apache Hive is an open source and collaborative project managed by the Apache Software Foundation and has started as a subproject of the Apache Hadoop. But due to its importance, it has its own scope today. It represents a Data Warehouse software, which facilitates the management, manipulation, and extraction of information from large sets of data [24, 25].

The Apache Hive and Apache Hadoop can be coupled to work together and provide users with distributed processing managed by the YARN, by using the MapReduce method and the HDFS distributed storage, without sacrificing ease and usability.

To abstract the complexity, the Apache Hive offers a query language named Hive Query Language (HiveQL), simple and similar to Structured Query Language (SQL), which is later converted into MapReduce jobs and run on the Apache Hadoop Cluster [24, 25].

Although Apache Hive is similar to Database Management Systems (DBMSs) such as: MySQL, PostgreSQL, Oracle, among others, it has high latency even when working with a small set of data. But this time interval is relatively small when it comes to a large data



set. For this reason, it is indicated to work with Big Data, where traditional DBMSs do not offer results in the expected time [24, 25].

Figure 8 illustrates the Apache Hive architecture working in conjunction with Apache Hadoop. The main components of the Apache Hive are: (1) User Interface (UI)—the User Interface to send requests to the system; (2) Driver—the Device that receives queries, implements the notion of session identifiers, provides execution, and searches APIs modeled on Java Database Connectivity (JDBC)/Open Database Connectivity (ODBC) interfaces; (3) Compiler—the Device that analyzes the HiveQL language, obtains the necessary metadata from Metastore and, eventually, generates an execution plan; (4) Metastore—the Device that stores the metadata, all the structure information of the various tables and partitions in the Warehouse; and, (5) Execution Engine—the Device that executes the execution plan created by the compiler [24, 25].

### The Optimized Row Columnar (ORC)

The ORC file format implements optimized columnar data storage and brings some benefits to the Apache Hive and Apache Hadoop. It uses data compression algorithms, reduces the storage space consumed, and brings performance-related advantages, as its columnar structure and division into data blocks allows a HiveQL query to access only the data blocks it needs, in addition to paralleling the reading data between multiple nodes and to increasing speed during query processing [25].

The ORC divides the data into blocks called stripes. These blocks are linear and, according to the standard configuration, have the size of 250 MB, considered as a good practice to obtain efficiency in the HDFS. Each stripe contains three fields: (1) index—for indexing the data for each block; (2) data—to store each block of data; and, finally, (3) information—to provide quick access to some calculations on the data of each relative column, such as: lowest value, highest value, average, sum, etc. These three elements are structured in columns, which represent each field in the database, and allow the query engines to read only the columns used in the query, for example, which optimizes access to the HDFS [25].

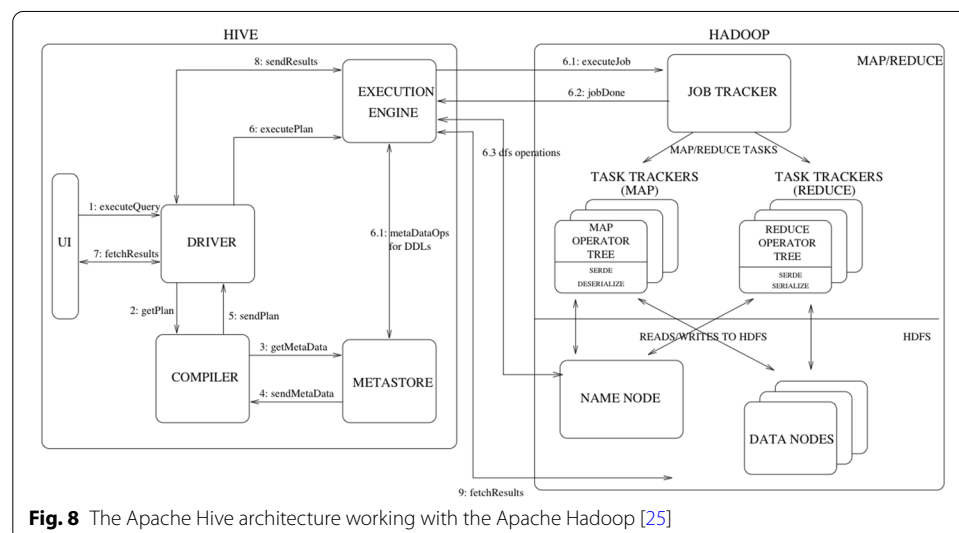




Figure 9 illustrates the architecture of the ORC and its various linear blocks, in addition to the internal structure of each block and the storage of the index, data, and information in column format.

### Related work

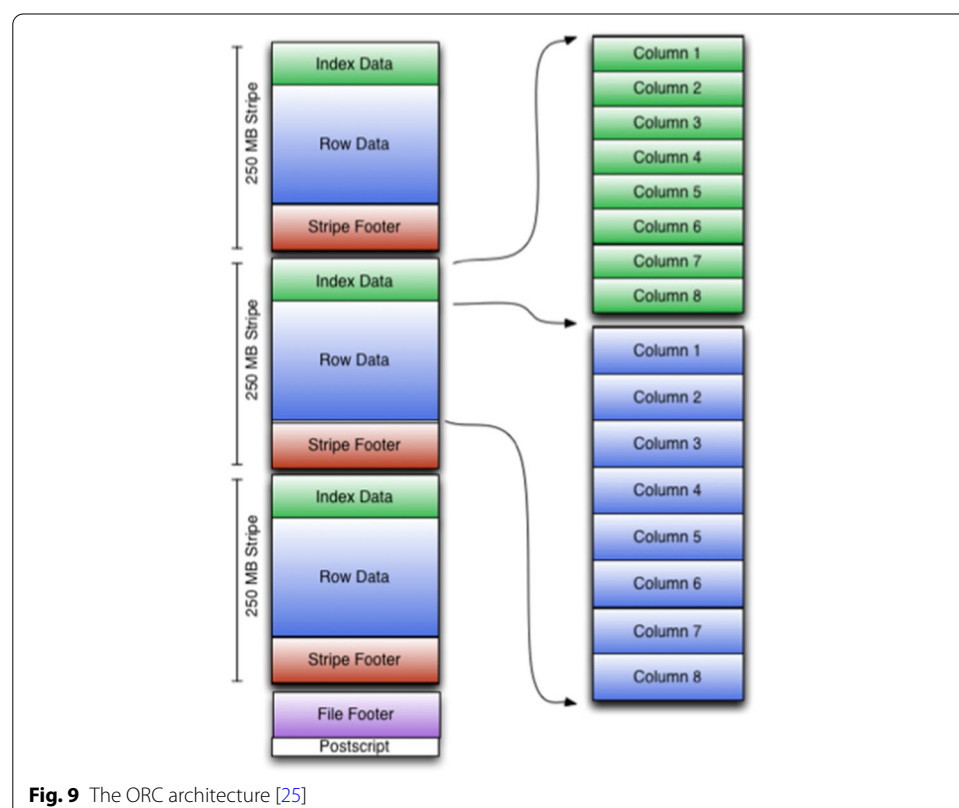
This section presents the main existing and related works to the scope of this research. It provides an overview of systems, architectures, tools used, types of data used, type of information generated, as well as their main limitations.

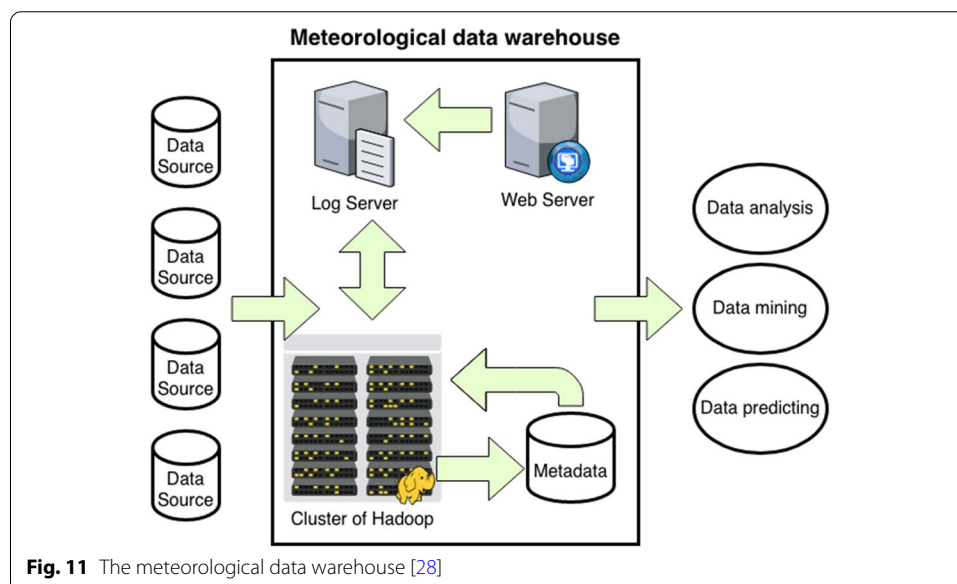
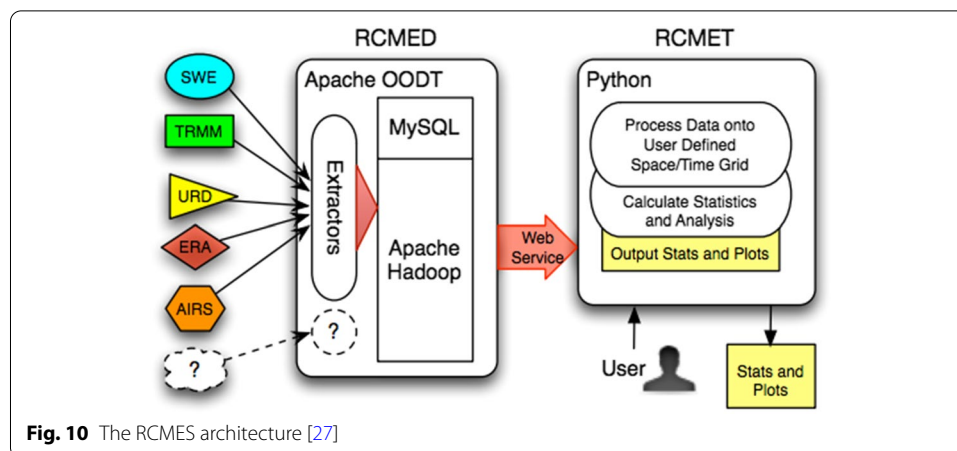
The existing systems were classified into three groups: (1) The use of the observed data; (2) The visualization of the weather/climate forecast; and (3) The storage and performance in the use of meteorological data.

Fathi et al. presented a systematic literature review of big data analytics in weather forecasting. Initially, 185 articles published between 2014 and August 2020 were identified. Then, 35 articles were chosen. Hadoop, Hive, Spark, Kafka, and Python were among the main technologies used to work with Big Data. Within this literature review, the main terms related to weather/climate forecast were: temperature with 29.7%, wind with 15.4%, precipitation with 11%, humidity with 12.1%, precipitation with 11% and pressure with 8.8% [26].

### The use of the observed data

Hart et al. have implemented a system called the Regional Climate Model Evaluation System (RCMES), which provides dynamic assessment of the regional climate model.





They compared the seasonal climate forecast between different models with observed data. Figure 10 illustrates the RCMES architecture, composed of two main modules: (1) The Regional Climate Model Evaluation Database (RCMED), which performs data ingestion and processing; and (2) The Regional Climate Model Evaluation Toolkit (RCMET), which integrates several tools for analyzing and visualizing information [27].

Shao et al. proposed a data mining environment in a meteorological data warehouse on a Hadoop cluster platform, shown in details in Fig. 11. The structure of the meteorological data warehouse is composed of a Hadoop cluster, a metadata storage system, a web server and a log server. This heterogeneous database is connected to other tools: data analysis, data mining, and data predicting [28].

Almgren et al. enabled the realization of the climate calculation on data observed in the last 63 years [29]. Waga and Rabah implemented a system based on Big Data and observed data to support agriculture [30]. Based on historical weather data, Wang

et al. proposed the creation of a tool to support decision making in the face of major climatic variations and natural disasters related to the energy system [31].

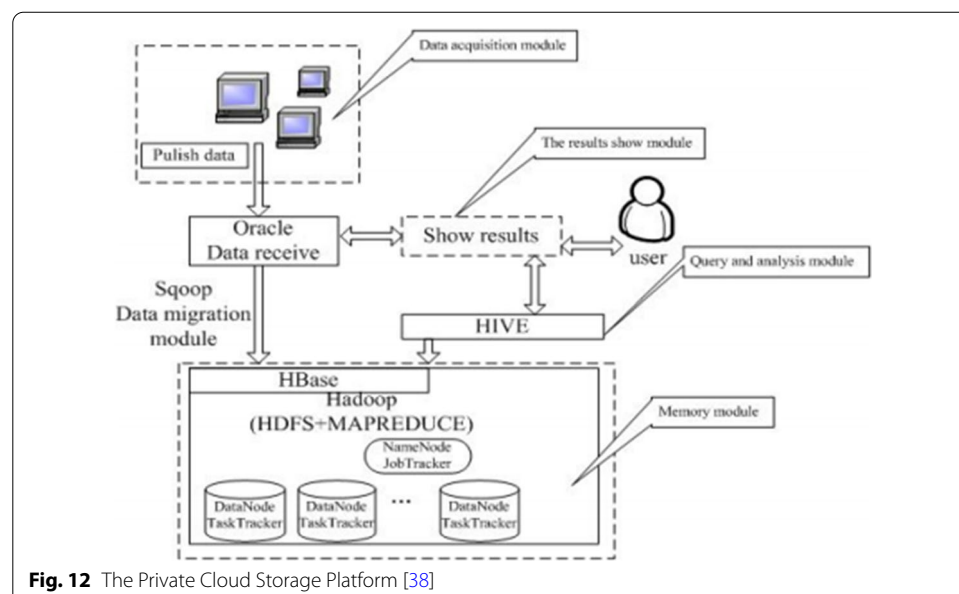
Chen et al. implemented the BigSmog system, which allowed the analysis of catastrophic conditions of severe pollution, using Big Data and historical records of data collection stations [32]. Yerva et al. have based on large volumes of observed meteorological data and social data to generate a concept about a target destination, this application is called Mood Space [33].

Mao and Zhu developed a Big Data application to support agriculture and micro-credit systems, based on calculations over a large volume of historical weather data, as well as a benchmark between a common computer and a Hadoop cluster [34].

Manogaran and Lopez stored a large volume of observed weather data in the HDFS database and used a MapReduce algorithm to make seasonal weather forecasts. Afterwards, they used a climate change detection algorithm based on spatial auto-correlation to monitor climate change [35].

### The visualization of the weather/climate forecast

Han and Yan created a cloud-based solution that allowed the use of mobile devices to view weather forecasts for cities, quickly and accurately [36]. Rutledge et al. explored a web interface and Big Data concepts to access meteorological data [37]. Finally, Xuelin et al. worked with a massive meteorological database, but with a focus on data management, storage, reading, and writing. Figure 12 illustrates the structure of the Private Cloud Storage Platform that has five modules: the Data Acquisition module, the Results Show module, the Query and analysis module, the Memory module, and the Data Migration module. The relational database was used as an intermediary to obtain greater performance in data insertion [38].



**Fig. 12** The Private Cloud Storage Platform [38]

### **The storage and performance in the use of meteorological data**

Bauer et al. presented a platform for storing, processing, and managing corporate data in a private cloud, where various types of data can be stored, such as: sales, marketing, weather and climate conditions, news and social media, among others [39].

Xie et al. addressed the challenge of Big Data in the storage and backup of images in JPEG format, where traditional techniques did not provide efficient data deduplication and proposed the use of the PXDedup technique. With the use of this technique, it was possible to unzip the JPEG file, divide it into fragments, eliminate visually identical fragments, and recompact the remaining parts before storage, thereby reducing considerably the size of the stored file, without considerable loss of quality [40].

With a focus on performance in the use of meteorological data, Fang et al. implemented a MapReduce version of the K-means algorithm called MK-means, thus overcoming the bottleneck in working with large sets of meteorological data [41]. Xue et al. performed a benchmark on a volume of meteorological data containing small files and then grouped into larger blocks [42]. Li et al. proposed an image processing technology to generate the weather forecast based on nephogram recognition, using a neural network method based on the k-means algorithm and a Hadoop cluster for performance processing [43].

According to the literature review that surrounds the themes of Big Data, Apache Hadoop, and Weather and Climate Forecast, the related works have explored the use of these subjects, but with emphasis on observed data, visualization of weather and climate forecast, and also on storage and performance in the use of meteorological data.

Preserving certain similarities, however different from these previous researches, this proposed model was named “A canonical model for seasonal climate prediction using Big Data” to perform seasonal climate predictions on the output data of the numerical climate forecast model, through its methods, processes, techniques, and tools of Big Data.

### **Describing and applying the canonical model architecture in a case study**

Initially, this section describes the challenge of this research when carrying out an operational seasonal climate prediction at CPTEC/INPE. To face this challenge, the seasonal climate prediction was divided into two stages: the Execution of the Global-CPTEC Model (parallel) and the Post-processing (sequential).

The first stage, the Execution of the Global-CPTEC Model is carried out in parallel, using resources from the Cray XE-6 supercomputer. The second stage, the Post-processing is sequential, involving data generated by the Global-CPTEC model. It was considered the focus of this research, mainly because it could be improved and optimized when using technologies associated with Big Data.

It is in this Post-processing stage that this research provides processing and distributed data storage, managed with simplicity and usability, using Big Data technologies such as: Hadoop Apache, Hive, YARN, MapReduce, HDFS, among others.

This section describes: (1) [The research challenge](#); (2) [The development of the proposed canonical conceptual model](#), its architecture in four modules, and its internal components; and (3) [The development of the proposed method for predicting seasonal](#)

climate from Big Data (M4PSC-BD) with its three internal processes. At the end of this section, it is also described the application of some metrics used for processing seasonal climate predictions.

### Describing the research challenge

The CPTEC/INPE uses the Global-CPTEC model to generate climate forecast products for Brazil. This model has a T062 resolution with an accuracy of approximately 200 km<sup>2</sup> and employs the prediction method by sets of members named Ensemble Method.

As described in “[The seasonal climate prediction](#)” section, it is necessary to subtract model’s climatology from the seasonal climate prediction to obtain the forecast of climatic anomalies, which allows, for example, to infer whether a given month or trimester will be more or less rainy than normal.

Model’s climatology refers to its ability to make predictions in certain regions or to indicate the climate behavior of these locations. It is obtained by running the model on a large set of historical data, which is why it is called long execution.

The current climatology of the Global-CPTEC model was generated from 1979 to 2010. This process takes 8 months to process and is only performed once per model. Processing is similar to that of seasonal climate prediction. However, it takes into account more variates with 2 processes of initial conditions and three variations of the model: the Persistent Sea Surface Temperature (in Portuguese: *Temperatura de Superfície do Mar—TSM*) with Kuo convection; the Relaxed Arakawa–Schubert (RAS); and the Grell.

The Persistent TSM consists of a procedure used in global climate modeling, where the atmospheric model receives, as one of its input parameters, the values of TSM anomalies observed from the month prior to the beginning of the forecast date and these values are maintained (persisted) throughout the model execution.

The TSM Prevista means that, instead of persisting the observed TSM anomalies, the TSM values for each forecast month are predicted by another auxiliary climate model. Therefore, in the first case, these input variables (TSM) are used at the beginning of the climate model execution, which are maintained throughout the processing. In the second case, an auxiliary model generates these variables for every model execution day.

The CPTEC/INPE uses the TSM forecasts of the Coupled Forecast System model version two (CFSv2) from the National Centers for Environmental Prediction (NCEP), as input to the global climate model. Its execution occurs in conjunction with the TSM conditions foreseen for the period that it is intended to forecast atmospheric conditions. Finally, the terms convection Kuo, RAS, and Grell are about three distinct ways of representing the process of cloud formation by convection (upward vertical movements) [11, 44–47].

In the seasonal climate prediction, the following 6 versions of the model are executed: the TSM persisted with Kuo, RAS, and Grell convection; and the TSM predicted with Kuo, RAS, and Grell convection. The model is applied for a period of 10 months, the first 3 months being retroactive and necessary for it to stabilize. The 4th month is the current month, the next 3 months are considered for the seasonal climate prediction, also called the target trimester, and the remaining interval is used for scientific research. Therefore, assuming that the current month is May, execution will be from February to November and the target trimester will be: June, July, and August.

Through the Ensemble forecasting technique (sets), the Global-CPTEC model is executed 15 times (members) with initial conditions of 15 different days. Therefore, for the climatology of the model, data are produced referring to: 3 variations of the model  $\times$  15 members  $\times$  90 days  $\times$  12 months  $\times$  30 years; and for the seasonal climate prediction, data are reproduced referring to: 6 variations of the model  $\times$  15 members  $\times$  90 days.

After the model execution, there is a post-processing stage to obtain the seasonal climate prediction. This stage consists of opening the binary files generated by the model and calculating, for example, the average or accumulation of variables for 15 daily members, months, and the target trimester. In this way, a single binary output file is generated with the result. The methodology used by the CPTEC/INPE makes use of FORTRAN's and Grad's tools. The target trimester has approximately 8190 files and, in this case, the total volume is around 24.570 MB.

The generated files are in binary format, encoded, and have several dimensions and levels. Each file can be represented as an XY matrix on the analyzed territory, where some variables have a value only at ground level, but other variables have different values, according to the height of the atmosphere (Z coordinate) and all data vary with the time.

The operational seasonal climate prediction at CPTEC/INPE is divided into two stages: the Execution of the Global-CPTEC model (parallel); and the Post-processing (sequential).

The first stage is performed on the Cray XE-6 supercomputer, which has a maximum processing capacity of 258 Teraflops as measured by the Linpack benchmark and its hardware configuration consists of 1304 nodes with 2 processors with 12 cores each, totaling 31,296 processors, approximately 40.75 TB of RAM, 866 TB of primary storage, 3.84 PB of secondary storage, and 6.0 PB of tertiary storage.

In the second stage of seasonal climate prediction, the data generated by the Global-CPTEC model is Post-processed. This stage is sequential and can be optimized. The use of the Apache Hadoop in conjunction with the Apache Hive has provided: the distributed processing managed by YARN; the use of the MapReduce method; and the HDFS distributed storage, in addition to ease and usability.

### **Describing the development of the proposed canonical conceptual model**

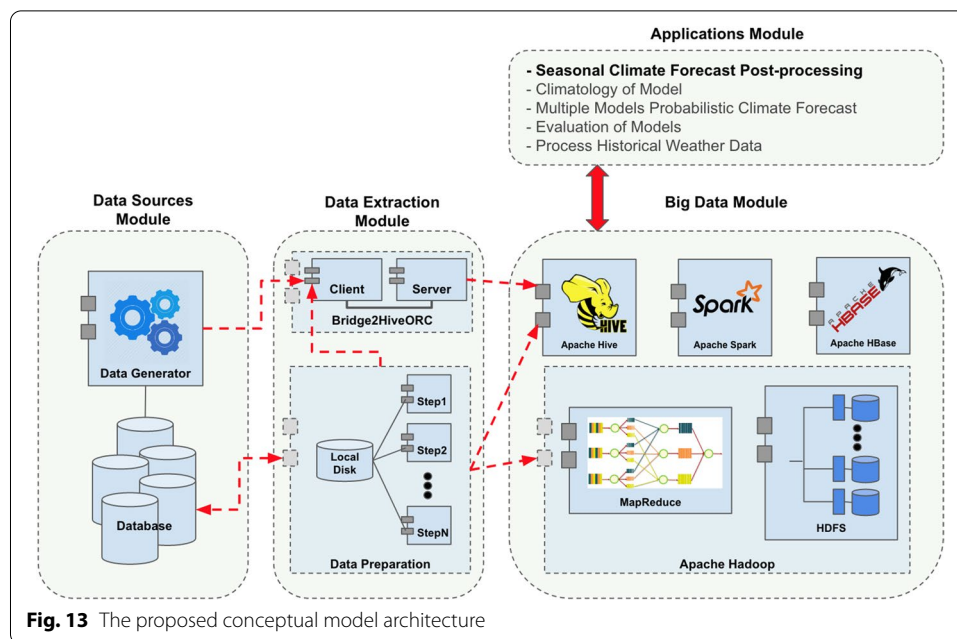
The challenges related to the five dimensions that define the term Big Data (Volume, Variety, Velocity, Value, and Veracity) have motivated the conception and development of a canonical model that integrated some Big Data tools and minimized the main difficulties related to handling large amounts of files and large data sets.

Figure 13 illustrates the proposed conceptual model architecture within its four modules: (1) [The Data Sources Module](#); (2) [The Data Extration Module](#); (3) [The Big Data Module](#); and (4) [The Applications Module](#).

#### ***The Data Sources Module***

The Data Sources Module was designed to represent several types of storage systems and the Data Generator Component. In this module, raw data can be directly extracted from the database or inserted in the Big Data environment, through the Data Generator Component.





### The Data Extraction Module

The Data Extraction Module was conceived to allow the insertion of data in the Big Data environment. This module consists of two components: (1) the Bridge2Hive-ORC Component; and (2) the Data Preparation Component.

The Bridge2HiveORC Component has a server subcomponent, which receives HiveQL commands through messages and executes them directly in the database offered by the Apache Hive component. The client forwards the received messages to the server.

The Data Preparation Component connects to the raw database of the Data Sources module and performs copying, processing, and standardization of data on local disk. After this step, this component also connects to the Apache Hive and the Apache Hadoop components of the Mass Data Repository module to transfer data to the Big Data environment.

### The Big Data Module

The Big Data Module was designed to centralize a large volume of structured and unstructured data and uses interconnected Big Data tools. It consists of four components: (1) the Apache Hadoop Component; (2) the Apache Hive Component; (3) the Apache Spark Component; and (4) the Apache HBase Component.

The Apache Hadoop Component ensures increased processing efficiency and scalability of all resources involved, which ensures greater speed of data capture, discover, and analysis. It has the HDFS and MapReduce subcomponents.

The Apache Hive Component promotes: greater efficiency in using the resources of the Apache Hadoop component; less complexity through the use of HiveQL queries; and greater dynamics and flexibility.

The Apache Spark Component, on the other hand, persists intermediate results in memory, instead of storing them on disk (HDFS), which generates greater efficiency to the Apache Hadoop component and can reach a speed up to 100 times faster.

Finally, the Apache HBase Component offers a distributed and scalable database, which supports the storage of structured data from large tables, allowing real-time and random access (read/write), as well as ease of integration with Hadoop.

### ***The Applications Module***

The Applications Module consists of a set of applications that make use of the Big Data Module. These applications have similar needs such as: extracting useful information from large data sets in a timely manner; subsequent MapReduce processing with writing to disk and memory; real-time and random access (read/write); among others.

### **Describing the development of the proposed method for predicting seasonal climate from Big Data (M4PSC-BD)**

The canonical model proposed in this research was applied in the post-processing of the seasonal climate prediction of the supercomputing environment of the Center for Weather Forecasting and Climate Studies linked to the Brazilian Institute for Space Research.

In order to perform a more appropriate processing, based on Big Data Sets, the Method for seasonal climate prediction was conceived, developed, and named in Portuguese *Método para PREdição CLImática Sazonal* resulting in the acronym M-PRECLIS.

The M-PRECLIS method consists of three processes: the Real Time Insertion Process, named in Portuguese *Processo de INSERção em TEmpo Real*, resulting in the acronym P-INSERTER; the Explicit Stored Data Extraction Process, named in Portuguese *Processo de EXTRAção expliCiTa de dados aRmazenados*, resulting in the acronym P-EXTRACTOR; and the seasonal climate prediction Process, named in Portuguese *Processo de PREvisão CLImática Sazonal*, resulting in the acronym P-PRECLIS.

Using the M-PRECLIS method, data can be obtained in two ways: from the development of the P-INSERTER process, involving the Bridge2HiveORC Component of the Data Extration Module and the Data Generator Component of the Data Source Module; or from the development of the P-EXTRACTOR process, involving the Data Preparation Component of the Data Extration Module and the Database Component of the Data Source Module. Thus, after an adequate preparation of the data, the following process of the seasonal climate prediction occur, based on the P-PRECLIS process, which performs the interaction between the Applications Module and the Big Data Module.

### ***Developing the process P-INSERTER***

The Real Time Insertion Process, named in Portuguese by the acronym P-INSERTER involved the relationship between the Bridge2HiveORC Component of the Data Extration Module and the Data Generator Component of the Data Source Module.

The BRIDGE to HIVE-ORC named in English by the acronym (Bridge-2HiveORC), was developed to simplify the data preparation process. The development of this component allowed data to be inserted directly into the HIVE database in Optimized Row Columnar (ORC) format. In this case, the Client and Server subcomponents were

applied, which performed a messaging service between an external application and the Big Data environment.

Thus, it was used the component previously described directly in the numerical model of climate forecast (coded in FORTRAN language). After being stored in a matrix in RAM memory, its data could then be inserted directly into the HIVE database already in ORC format, using the DML INSERT command in the HiveQL language.

Although this component is more optimized, it was not always trivial to change a complex application. For this reason, the proposed model has these two forms of data extraction.

### ***Developing the process P-EXTRACTOR***

The Explicit Stored Data Extraction Process, named in Portuguese by the acronym P-EXTRACTOR involved the relationship between the Data Preparation Component of the Data Extraction Module and the Database Component of the Data Source Module, consisting of the following seven steps: (1) Generate the List of Files; (2) Transfer Files; (3) Convert Binary to Text; (4) Create the Apache Hive Database Table; (5) Transfer the Converted Text Format Data into the Hive Database Format; (6) Create the Apache Hive Database Table using the ORC format; and (7) Convert the Apache Hive Database into the ORC Format.

**Step 1—Generate the list of files:** The first step of P-EXTRACTOR consisted of locating and filtering the binary output files of the Global-CPTEC numerical model, present in the supercomputing environment, referring to the target trimester and eliminating unnecessary data.

**Step 2—Transfer files:** This step of P-EXTRACTOR aimed to transfer the binary files of the target trimester, from the supercomputing environment, to the Big Data processing cluster.

**Step 3—Convert binary to text:** In this step of P-EXTRACTOR, the binary files were converted to text format, ideal for greater efficiency in Hadoop, as it involves processing and distributed data storage.

These binary files had four dimensions: X (longitude), Y (latitude), Z (altitude), and T (time). The X dimension had 192 points of longitude, the Y dimension had 96 points of latitude, the Z dimension had up to six vertical pressure levels (1,000, 925, 850, 500, 250 and 200 mbars), and the T dimension varied daily in the period of reference.

Each XYZT point had 13 types of variables separated into two groups. The first group, containing seven variables, had value only at one level: TOPO, LSMK, T02M, TSZW, TSMW, SPMT, and PREC. The second group consisted of six variables at six levels: UVMT, VVMT, GHMT, TMAT, UEMT, and OMMT.

The final file contained only two dimensions. One dimension with an index ranging from 1 to 18,432 and the other dimension with the following 48 variables: INDEX, YEAR, MONTH, DAY, MODEL, TOPO, LSMK, T02M, TSZW, UVMT1, UVMT2, UVMT3, UVMT4, UVMT5, UVMT6, TSMW, VVMT1, VVMT2, VVMT3, VVMT4, VVMT5, VVMT6, GHMT1, GHMT2, GHMT3, GHMT4, GHMT5, GHMT6, SPMT, TMAT1, TMAT2, TMAT3, TMAT4, TMAT5, TMAT6, UEMT1, UEMT2, UEMT3,

UEMT4, UEMT5, UEMT6, OMMT1, OMMT2, OMMT3, OMMT4, OMMT5, OMMT6, and PREC.

Step 4—Create the Apache Hive Database Table: In this step of P-EXTRACTOR, the table was created in the Apache Hive Database. This table had exactly the same fields as the text file shown in the previous step and kept the same order of its elements.

The fields “cli\_index”, “cli\_year”, “cli\_month”, and “cli\_day” were created of type integer (INT), the field “cli\_model” was configured as a string (STRING), and the remaining 43 variables were considered floating type (FLOAT).

The command in Data Definition Language (DDL) used to define this data structure is shown in Fig. 14.

Step 5—Transfer the converted text format data into the Hive Database format: This fifth step of P-EXTRACTOR dealt with the transfer of data converted from textual format to the format of the HDFS file system. Before, it was necessary to create the directory specified in the “LOCATION” parameter of the DDL command seen in the previous step. This was done using the “hdfs dfs -mkdir /MODELDATA” command. Then, the data was transferred to the HDFS, using the “hdfs dfs -put <FILE>/MODELDATA” command. As for the disk volume, contents of files already converted to text format have used 59.4 GB of space, increasing by 2.48 times the original size, which was 24 GB.

Step 6—Create the Apache Hive database table using the ORC format: In this sixth step of the P-EXTRACTOR, a table similar to the table from step 4 was created, changing the format of the Database stored in the Hive to the Optimized Row Columnar (ORC) format.

The ORC format offers to Hive more efficiency if compared to the purely textual format. It manages to improve performance, due to the restructuring of the data in linear groups named “stripes”. In addition to this, the other advantage is in data compression, significantly reducing disk usage.

The DDL command used to define this data structure is shown in Fig. 15 and the “STORED AS ORC” setting indicates the storage format.

```
CREATE TABLE climate (
  cli_index INT, cli_year INT, cli_month INT, cli_day INT, cli_model STRING,
  cli_topo FLOAT, cli_lsmk FLOAT, cli_t02m FLOAT, cli_tszw FLOAT,
  cli_uvmt1 FLOAT, cli_uvmt2 FLOAT, cli_uvmt3 FLOAT, cli_uvmt4 FLOAT,
  cli_uvmt5 FLOAT, cli_uvmt6 FLOAT, cli_tsmw FLOAT, cli_vvmt1 FLOAT,
  cli_vvmt2 FLOAT, cli_vvmt3 FLOAT, cli_vvmt4 FLOAT, cli_vvmt5 FLOAT,
  cli_vvmt6 FLOAT, cli_ghmt1 FLOAT, cli_ghmt2 FLOAT, cli_ghmt3 FLOAT,
  cli_ghmt4 FLOAT, cli_ghmt5 FLOAT, cli_ghmt6 FLOAT, cli_spm1 FLOAT,
  cli_tmat1 FLOAT, cli_tmat2 FLOAT, cli_tmat3 FLOAT, cli_tmat4 FLOAT,
  cli_tmat5 FLOAT, cli_tmat6 FLOAT, cli_uemt1 FLOAT, cli_uemt2 FLOAT,
  cli_uemt3 FLOAT, cli_uemt4 FLOAT, cli_uemt5 FLOAT, cli_uemt6 FLOAT,
  cli_ommt1 FLOAT, cli_ommt2 FLOAT, cli_ommt3 FLOAT, cli_ommt4 FLOAT,
  cli_ommt5 FLOAT, cli_ommt6 FLOAT, cli_prec FLOAT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ' '
LOCATION '/MODELDATA/';
```

**Fig. 14** The DDL command code used to create the Hive Table

```

CREATE TABLE climate_orc (
  cli_index INT, cli_year INT, cli_month INT, cli_day INT, cli_model STRING,
  cli_topo FLOAT, cli_lsmk FLOAT, cli_t02m FLOAT, cli_tszw FLOAT,
  cli_uvmt1 FLOAT, cli_uvmt2 FLOAT, cli_uvmt3 FLOAT, cli_uvmt4 FLOAT,
  cli_uvmt5 FLOAT, cli_uvmt6 FLOAT, cli_tsmw FLOAT, cli_vvmt1 FLOAT,
  cli_vvmt2 FLOAT, cli_vvmt3 FLOAT, cli_vvmt4 FLOAT, cli_vvmt5 FLOAT,
  cli_vvmt6 FLOAT, cli_ghmt1 FLOAT, cli_ghmt2 FLOAT, cli_ghmt3 FLOAT,
  cli_ghmt4 FLOAT, cli_ghmt5 FLOAT, cli_ghmt6 FLOAT, cli_spm1 FLOAT,
  cli_tmat1 FLOAT, cli_tmat2 FLOAT, cli_tmat3 FLOAT, cli_tmat4 FLOAT,
  cli_tmat5 FLOAT, cli_tmat6 FLOAT, cli_uemt1 FLOAT, cli_uemt2 FLOAT,
  cli_uemt3 FLOAT, cli_uemt4 FLOAT, cli_uemt5 FLOAT, cli_uemt6 FLOAT,
  cli_ommt1 FLOAT, cli_ommt2 FLOAT, cli_ommt3 FLOAT, cli_ommt4 FLOAT,
  cli_ommt5 FLOAT, cli_ommt6 FLOAT, cli_prec FLOAT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ' '
STORED AS ORC
LOCATION '/MODELDATA-ORC/';

```

**Fig. 15** The DDL code to create the Hive Table in ORC format

Step 7—Convert the Apache Hive Database into the ORC format: In this seventh and final step of the P-EXTRACTOR, a Data Manipulation Language (DML) command was used to complete the conversion from the Text format to the ORC format. Initially, it was necessary to create the MODELDATA-ORC directory, using the “hdfs dfs -mkdir /MODELDATA-ORC” command. Then, the conversion was carried out using the “hive -S -e “INSERT INTO TABLE climate\_orc SELECT \* FROM climate”” command.

After converting to the ORC format, the disk space used was 19.7 GB, decreasing 1.22 times if compared to the binary format (24 GB), and 3.02 times if compared to the text format (59.4 GB).

At the end of the application of these seven steps of the proposed P-EXTRACTOR process, the gain obtained with the significant reduction in storage space was evident.

#### **Developing the process P-PRECLIS**

The seasonal climate prediction process, named in Portuguese by the acronym P-PRECLIS, has involved the relationship between the Applications Module and the Big Data Module.

The processing of the seasonal climate prediction was obtained by executing a HiveQL query, which encapsulated metrics to measure the complexity of Hadoop (MapReduce and HDFS) and Hive in the ORC format.

Equation 1 presents an example of application of these metrics used for seasonal climate prediction of average precipitation, based on the following three subsequent average calculations. First, the average of the 15 Ensemble members generated per day was calculated. Then, the monthly average was calculated and, finally, the trimesterly average.

In the first part of the calculation,  $\frac{1}{p} \cdot \sum_{i=1}^p x_i$ ,  $p$  represented the number of Ensemble members (15). In the following calculation,  $\frac{1}{d} \cdot \sum_{j=1}^d y_j$ ,  $d$  represented the number of daily averages in the month (30, for example). And, finally, in the calculation  $\frac{1}{s} \cdot \sum_{k=1}^s z_k$ ,  $s$  represented the number of monthly averages (3, for example).

This calculation corresponded to approximately 151,142,400 tuples (tuples of each file multiplied by the number of files). These tuples were grouped by the value of the

index, which ranged from 1 to 18,432. As each tuple contained 48 variables, an approximate total of 7.25 billion elements was analyzed at the end (total tuples multiplied by the number of fields).

$$Average = \left[ \frac{1}{s} \cdot \sum_{k=1}^s z_k \left[ \frac{1}{d} \cdot \sum_{j=1}^d y_j \left[ \frac{1}{p} \cdot \sum_{i=1}^p x_i \right] \right] \right] \quad (1)$$

A code fragment in Python is illustrated in Fig. 16, which presents as a result, the index and the average precipitation of the seasonal climate prediction obtained, where the grouping and order were applied to the index field. With this, it was found that it would also be possible to perform other types of calculations while maintaining the same calculation methodology.

### The main results and discussions

The experiment of this research was executed in a Hadoop Cluster and involved a front-end equipment and 32 slave nodes. In this experiment, distributed storage resources from the HDFS storage system and parallel processing were used through the MapReduce functionalities.

The front-end equipment was configured with 4 processing cores, 16 GB of RAM, 4 disks with 73 GB, and an additional disk with 147 GB of local storage. Each slave node was configured with 4 cores, 8 GB of RAM, and 1 disk with 250 GB of local storage. On this disk, a partition with 184 GB of storage was reserved for HDFS, which was grouped in the HDFS system with other partitions, totaling 5.7 TB of storage.

The first stage of verification of the proposed canonical model has shown precision in the results produced. The proposed seasonal climate prediction method—M-PRECLIS was applied in the processing of average precipitation resulting in six different forecast versions: the TSM persisted with Kuo, RAS, and Grell convection; and the TSM expected with Kuo, RAS, and Grell convection.

These files were compared with the operational files and were considered identical. Figure 17 shows an example using output files plotted in the Grads tool.

In this example, it was found that the code implemented was of low cyclomatic complexity, as it has few indented instructions and a small number of operators and operands. Thus, using the Halstead and Cyclomatic Complexity metrics would not result in significant values for analysis. However, this fact proved that the code was testable, of low complexity, and easy to maintain.

```
import struct

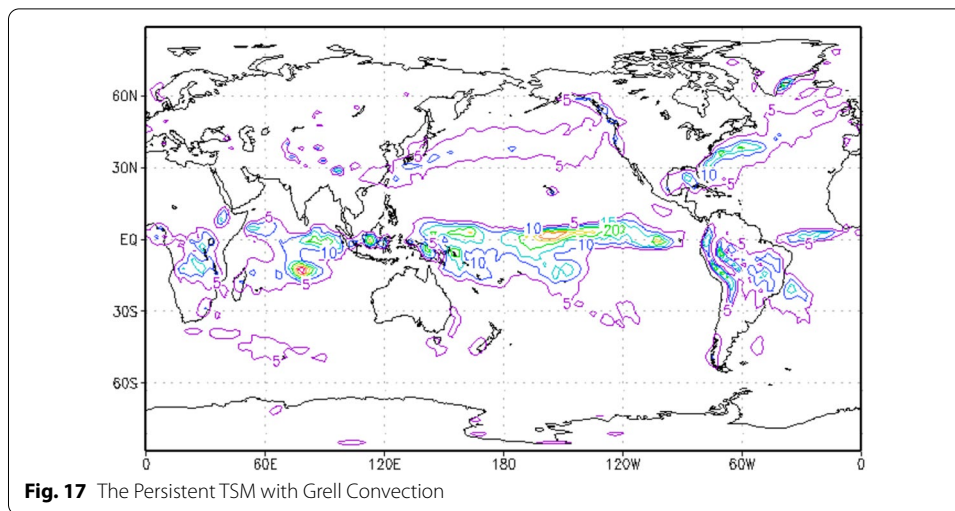
# Process: P-PRECLIS
cmd="hive -S -e 'select cli_index, cli_model, AVG(cli_prec) as cli_prec from (select cli_index, cli_year, cli_month, cli_model, AVG(cli_prec) as cli_prec from (select cli_index, cli_year, cli_month, cli_model, AVG(cli_prec) as cli_prec from c lima_orc group by cli_index, cli_year, cli_month, dia, cli_model) as s1 group by cli_index, cli_year, cli_month, cli_model) as s2 group by cli_index, cli_model order by cli_model, cli_index' > selectall"
status,output=commands.getstatusoutput(cmd)

# Convert the output to binary format
for variation in ['grellpersisted', 'grellforecasted', 'kuopersisted', 'kuoforecasted', 'raspersisted', 'rasforecasted']:

    # Opens the output file for writing in blocks
    FILE="output.hadoop."+variation
    print FILE
    f = open(FILE, "w+b")
```

**Fig. 16** A Python language fragment of code for a seasonal climate prediction





Therefore, the metrics of source code lines and computational performance were chosen because they are more related to the characteristics of the implemented code.

As for the metrics of source code lines, only the third step of the Data Preparation Component used Python language code with 37 useful lines, the remaining steps were performed by line commands in the terminal. The use of the Bridge2Hive-ORC Component employed the DML INSERT command that belongs to the HiveQL language. The proposed seasonal climate prediction process P-PRECLIS was used based on a Python script with only 23 useful lines.

The execution duration of the steps referring to the *Data Preparation* Component was 30 min and 16 s and the data extraction occurred through the *Bridge2HiveORC* Component, performed in parallel with the generation of the binary files. Therefore, times were canceled.

As for performance, the use of M-PRECLIS, as a proposed method to carry out the processing of the seasonal climate prediction, consumed only 4 min and 8 s, with a maximum consumption of 60% of all cores, 60 GB of RAM, and 100 GB of cache.

According to [6], in a relevant observation related to mono-processed architecture, when it comes to Big Data, it is not possible to base calculations on the optimal parallelization equation, where the cost has the same order of sequential processing:  $p \cdot T_p = \theta(T_s)$ , because the hard drive becomes a bottleneck.

According to the bibliographic research carried out, the related works have an emphasis only on observed data, visualization of weather forecast, and climate and performance in the use of meteorological data.

Preserving certain similarities, but differently from these researches, the M-PRECLIS proposed and presented in this research, different from the traditional models and methods previously used, provided the most effective realization of a seasonal climate prediction on the output data of the Global-CPTEC numerical model, using: the Apache Hadoop framework, the MapReduce distributed processing, the HDFS distributed storage, the Apache Hive module, and the ORC file system.

## Conclusion

This article described the development of “A canonical model for seasonal climate prediction using Big Data”. For the development of the proposed model, the Method for Seasonal Climate Prediction was conceived and developed, named in Portuguese *Método para PREdição CLImática Sazonal* resulting in the acronym M-PRECLIS.

The proposed method M-PRECLIS was implemented in the experiment of a case study on Seasonal Climate Prediction, using the main methods, techniques, tools, and metrics of free software available in the market and applying Big Data technologies.

At the end of the experiment, it was found that the proposed model is consistent and produces correct results, in addition to having low code complexity and good performance. According to the literature review, it was identified that the proposed model is innovative when performing seasonal climate prediction on the output data of the atmospheric numerical model of climate area. Thus, the use of the model and the proposed method can provide greater dynamics, speed, and conciseness in Seasonal Climate Prediction.

In addition, from this experiment, it was also found that the proposed method has provided an environment capable of: integrating Big Data Sets and various types of meteorological data; centralizing data storage; avoiding raw data transfers and file replications; and at the same time, providing better scalability.

The authors of this article believe that the main contribution of this research work was the elaboration of the canonical model for seasonal climate prediction, based on Big Data.

The first additional contribution of this research carried out from the Canonical Model was the design and development of the seasonal climate prediction method named M-PRECLIS.

The second additional contribution of this research work was the use of: Python programming language; Apache Hadoop; Apache Hive; and the format of ORC files in the knowledge domain of Seasonal Climate Prediction.

The third additional contribution of this research was the analysis of the Canonical model application, the M-PRECLIS method and its three internal processes (P-INSERTER, P-EXTRACTOR, and P-PRECLIS) in a practical and real experiment, involving operational data from the seasonal climate prediction.

The authors of this article also consider as a complementary contribution of this research work, the satisfactory use of the proposed canonical model, mainly for providing faster calculations on a base, involving Big Data Sets of Meteorological Data, coming from a very robust Numerical Model of Climatic Forecasting.

On this context, in addition to providing a computational environment capable of integrating various types of meteorological data, centralizing data storage and increasing scalability, the proposed canonical model avoided raw data transfers and unnecessary file replications, saving several resources involved.

Therefore, based on the experiment carried out and from data presented in this article, it was found that the proposed canonical model constitutes a viable alternative that can guide new paths to be taken that, certainly, will demand new research and future improvements to corroborate the development of increasingly robust systems with different additional features.

The authors believe that the future of this important area of research will increasingly involve Large Meteorological Data Sets focusing on Big Data as a solution and will naturally evolve towards the creation and improvement of new models, methods, techniques, metrics, and tools for emerging computer systems.

For future work, it is suggested the use of data from other Numerical Models of Climate Forecasting, composing multi-model databases that allow even more accurate calculations of probabilistic climate predictions.

It is also suggested for future work to use observed historical meteorological data and historical series of model executions, with the purpose of producing statistics, supporting reliable analysis of models, and identifying evolutions or failures.

Finally, it is suggested the continuation of this research, addressing new case studies and even more complete and larger experiments involving Big Data Sets, new models, methods, techniques, metrics, and tools for the development of even more agile and robust computer systems.

Among the main tools for deepening new research and its applications in the domain of knowledge of Seasonal Climate Prediction, it is suggested the use of the following Big Data technologies: Pig, as a scripting language for MapReduce; Hbase, as a Hadoop database; Flume, as a log export system; Sqoop, as a system for exporting DBMS data to Hadoop; Apache Cassandra, to provide linear scalability in NoSQL Databases; and/or Apache Spark, as a Framework capable of providing Data Analytics and real-time Machine Learning in distributed computing environments.

#### Abbreviations

CPTEC: Center for Weather Forecasting and Climate Studies (in Portuguese: *Centro de Previsão de Tempo e Estudos Climáticos*); INPE: Brazilian Institute for Space Research (in Portuguese: *Instituto Nacional de Pesquisas Espaciais*); IoT: Internet of Things; IT: Information Technology; YARN: Yet Another Resource Negotiator; HDFS: Hadoop Distributed File System; I/O: Input/Output; HiveQL: Hive Query Language; SQL: Structured Query Language; DBMSs: Database Management Systems; UI: User Interface; JDBC: Java Database Connectivity; ODBC: Open Database Connectivity; ORC: Optimized Row Columnar; RCMES: Regional Climate Model Evaluation System; TSM: Sea Surface Temperature (in Portuguese: *Temperatura de Superfície do Mar*); RAS: Relaxed Arakawa–Schubert; CFSv2: Coupled Forecast System model version two; NCEP: National Centers for Environmental Prediction; M4PSC-BD: Method for Predicting Seasonal Climate from Big Data; M-PRECLIS: Method for seasonal climate prediction (in Portuguese: *Método para PREdição CLImática Sazonal*); P-INSERTER: Real Time Insertion Process (in Portuguese: *Processo de INSERção em Tempo Real*); P-EXTRACTOR: Explicit Stored Data Extraction Process (in Portuguese: *Processo de EXTRAção explícita de dados armazenados*); P-PRECLIS: Seasonal climate prediction process (in Portuguese: *Processo de PREvisão CLImática Sazonal*); DDL: Data Definition Language; DML: Data Manipulation Language.

#### Acknowledgements

The authors thank: the Center for Weather Forecasting and Climate Studies linked to the Brazilian Institute for Space Research (*Centro de Previsão de Tempo e Estudos Climáticos—CPTEC do Instituto Nacional de Pesquisas Espaciais—INPE*); the Brazilian Aeronautics Institute of Technology (*Instituto Tecnológico de Aeronáutica—ITA*); the Casimiro Montenegro Filho Foundation (*Fundação Casimiro Montenegro Filho—FCMF*); and the Brazilian Enterprise *Ecosistema Negócios Digitais Ltda* for their support and infrastructure, which motivate the challenges and innovations of this research project.

#### Authors' contributions

All authors made substantial contributions. All authors read and approved the final manuscript.

#### Funding

The Casimiro Montenegro Filho Foundation (*Fundação Casimiro Montenegro Filho—FCMF*) and the Brazilian Enterprise *Ecosistema Negócios Digitais Ltda*.

#### Availability of data and materials

Not applicable.

#### Declarations

##### Ethics approval and consent to participate

Not applicable.

**Consent for publication**

Not applicable.

**Competing interests**

The authors declare that they have no competing interests.

**Author details**

<sup>1</sup>Computer and Electronic Engineering Graduate Program, Brazilian Aeronautics Institute of Technology, São José dos Campos, São Paulo, Brazil. <sup>2</sup>Center for Weather Forecasting and Climate Studies, Brazilian Institute for Space Research, Cachoeira Paulista, São Paulo, Brazil.

Received: 4 October 2021 Accepted: 15 February 2022

Published online: 03 March 2022

**References**

- Ylijoki O, Porras J. Perspectives to definition of big data: a mapping study and discussion. *J Innov Manag.* 2016;4(1):69–91. [https://doi.org/10.24840/2183-0606\\_004.001\\_0006](https://doi.org/10.24840/2183-0606_004.001_0006).
- Laney D. 3d data management: controlling data volume, velocity and variety. *META Group Res Note.* 2001;6(1):70.
- Gantz J, Reinsel D. Extracting value from chaos, 2011. <http://www.kushima.org/wp-content/uploads/2013/05/DigitalUniverse2011.pdf>. Accessed 17 Jan 2022.
- Schroeck M, Shockley R, Smart J, Romero-Morales D, Tufano P. Analytics: the real-world use of big data: how innovative enterprises extract value from uncertain data, executive report. IBM Institute for Business Value and Said Business School at the University of Oxford. 2012.
- Kaisler S, Armour F, Espinosa JA, Money W. Big data: issues and challenges moving forward. In: *IEEE, 46th Hawaii international conference on system sciences.* 2013. <https://doi.org/10.1109/HICSS.2013.645>.
- Ramos MP, Tasinaffo PM, Almeida ES, Achite LM, Cunha AM, Dias LAV. Distributed systems performance for big data. In: *Information Technology: new generations (ITNG), 2016 tenth international conference on,* 2016. p. 733–44. [https://doi.org/10.1007/978-3-319-32467-8\\_64](https://doi.org/10.1007/978-3-319-32467-8_64).
- Cavalcanti IFA, Ferreira NJ, Silva MGAJ, Dias MAFS. *Tempo e Clima No Brasil.* 1st ed. São Paulo: Oficina de textos; 2009. p. 463.
- Coelho CAS. Forecast calibration and combination: Bayesian assimilation of seasonal climate predictions. *Doutorado em meteorologia, University of Reading;* 2005.
- Chan CS. Previsões Climáticas Sazonais Geradas Pelo Modelo Eta do CPTEC/INPE. Instituto Nacional de Pesquisas Espaciais—INPE, Rod. Presidente Dutra, KM40, Cachoeira Paulista—SP. 2011. Instituto Nacional de Pesquisas Espaciais—INPE.
- Cavalcanti IFA, Marengo JA, Satyamurty P, Nobre CA, Trosnikov I, Bonatti JP, Manzi AO, Tarasova T, Pezzi LP, Almeida CD, Sampaio G, Castro CAC, Sanches MB, Camargo H. Global climatological features in a simulation using the cpctec-cola agcm. *J Clim.* 2002;15(21):2965–88. [https://doi.org/10.1175/1520-0442\(2002\)015%3C2965:GCFIAS%3E2.0.CO;2](https://doi.org/10.1175/1520-0442(2002)015%3C2965:GCFIAS%3E2.0.CO;2).
- Coelho CAS, Cavalcanti IFA, Costa SMS, Freitas SR, Ito SR, Luz G, Santos AF, Nobre CA, Marengo JA, Pezza AB. Climate diagnostics of three major drought events in the amazon and illustrations of their seasonal precipitation predictions. *Meteorol Appl.* 2012;19(2):237–55. <https://doi.org/10.1002/met.1324>.
- Machado RD, Rocha RP. Previsões climáticas sazonais sobre o brasil: avaliação do regcm3 aninhado no modelo global cpctec/cola. *Rev Brasil Meteorol.* 2011;26(1):121–36. <https://doi.org/10.1590/S0102-77862011000100011>.
- Marengo JA, Cavalcanti IFA, Satyamurty P, Trosnikov I, Nobre CA, Bonatti JP, Camargo H, Sampaio G, Sanches MB, Manzi AO, Castro CAC, Almeida CD, Pezzi LP, Candido L. Assessment of regional seasonal rainfall predictability using the cpctec/cola atmospheric gcm. *Clim Dyn.* 2003;21(5–6):459–75. <https://doi.org/10.1007/s00382-003-0346-0>.
- Huang L, Leng H, Li X, Ren K, Song J, Wang D. A data-driven method for hybrid data assimilation with multilayer perceptron. *Big Data Res.* 2021;23(1):1–18. <https://doi.org/10.1016/j.bdr.2020.100179>.
- Lorenz EN. Deterministic non-periodic flow. *J Atmos Sci.* 1963;20:130–41.
- Lorenz EN. A study of the predictability of a 28-variable atmospheric model. *Tellus.* 1965;17:321–33.
- Lorenz EN. The predictability of a flow which possesses many scales of motion. *Tellus.* 1969;21:289–307.
- Foundation AS. *APACHE-HADOOP.* 2020. <http://hadoop.apache.org/>. Accessed 17 Jan 2022.
- White T. *Hadoop: the definitive guide.* 3rd ed. California: O'Reilly Media Inc; 2012. p. 688.
- Shvachko K, Kuang H, Radia S, Chansler R. The hadoop distributed file system. In: *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th symposium on,* 2010. p. 1–10. <https://doi.org/10.1109/MSST.2010.5496972>.
- Pandey S, Tokekar V. Prominence of mapreduce in big data processing. In: *Communication Systems and Network Technologies (CSNT), 2014 fourth international conference on,* 2014. p. 555–60. <https://doi.org/10.1109/CSNT.2014.117>.
- Belcastro L, Cantini R, Marozzo F, Orsino A, Talia D, Trunfio P. Programming big data analysis: principles and solutions. *J Big Data.* 2022;9(1):1–50. <https://doi.org/10.1186/s40537-021-00555-2>.
- Pajoo HH, Rashid MA, Alam F, Demidenko S. lot big data provenance scheme using blockchain on hadoop ecosystem. *J Big Data.* 2021;8(1):1–26. <https://doi.org/10.1186/s40537-021-00505-y>.
- Foundation AS. *HIVE.* 2020. <https://hive.apache.org/>. Accessed 17 Jan 2022.
- Foundation AS. *APACHE-HIVE.* 2020. <https://cwiki.apache.org/confluence/display/Hive/Tutorial>. Accessed 17 Jan 2022.
- Fathi M, Kashani MH, Jameii SM, Mahdipour E. Big data analytics in weather forecasting: a systematic review. *Arch Comput Methods Eng.* 2021. <https://doi.org/10.1007/s11831-021-09616-4>.

27. Hart AF, Goodale CE, Mattmann CA, Zimdars P, Crichton D, Lean P, Kim J, Waliser D. A cloud-enabled regional climate model evaluation system. In: Proceedings of the 2nd international workshop on software engineering for cloud computing. 2011. p. 43–9. <https://doi.org/10.1145/1985500.1985508>.
28. Shao L, Liu J, Dong G, Mu Y, Guo P. The establishment and data mining of meteorological data warehouse. In: Mechatronics and automation (ICMA), 2014 IEEE international conference on, 2014. p. 2049–54. <https://doi.org/10.1109/ICMA.2014.6886019>.
29. Almgren K, Alshahrani S, Lee J. Weather data analysis using hadoop to mitigate event planning disasters. Bridgeport: University of Bridgeport Scholar Works; 2015.
30. Waga D, Rabah K. Environmental conditions? big data management and cloud computing analytics for sustainable agriculture. *World J Comput Appl Technol*. 2014;2(3):73–81.
31. Wang YF, Deng MH, Bao YK, Zhang H, Chen JY, Qian J, Guo CX. Power system disaster-mitigating dispatch platform based on big data. In: Power System Technology (POWERCON), 2014 international conference on, 2014. p. 1014–9. <https://doi.org/10.1109/POWERCON.2014.6993940>.
32. Chen J, Chen H, Pan JZ, Wu W, Zhang N, Zheng G. When big data meets big smog: a big spatio-temporal data framework for china severe smog analysis. In: Proceedings of the 2nd ACM SIGSPATIAL international workshop on analytics for big geospatial data. 2013. p. 13–22. <https://doi.org/10.1145/2534921.2534924>.
33. Yerva SR, Jeung H, Aberer K. Cloud based social and sensor data fusion. In: Information Fusion (FUSION), 2012 15th international conference on, 2012. p. 2494–501.
34. Mao H, Zhu L. The application of hadoop in natural risk prevention and control of rural microcredit. *Am J Ind Bus Manag*. 2015;3(03):102. <https://doi.org/10.4236/ajibm.2015.53011>.
35. Manogaran G, Lopez D. Spatial cumulative sum algorithm with big data analytics for climate change detection. *Comput Electr Eng*. 2018;65:207–21. <https://doi.org/10.1016/j.compeleceng.2017.04.006>.
36. Han X, Yan J. Application research of weather forecast query system based on cloud computing. *IJACT: Int J Adv Comput Technol*. 2013;5(1):722–32.
37. Rutledge G, Crichton D, Alpert J. Improving numerical weather prediction models and data-access latencies. *Earthzine*, March, 2014. p. 29.
38. Xuelin L, Junfeng X, Jiefang B. Research on private cloud storage solutions for meteorological applications. *DATA MINING PROVID PERSONAL LEARN MATER INTERACT*. 2014;2(1):41.
39. Bauer D, Froese F, Garces-Erice L, Giblin C, Labbi A, Nagy ZA, Pardon N, Rooney S, Urbanetz P, Vetsch P, Wespi A. Building and operating a large-scale enterprise data analytics platform. *Big Data Res*. 2021;23(1):1–20. <https://doi.org/10.1016/j.bdr.2020.100181>.
40. Xie H, Deng Y, Feng H, Si L. Pxdedup: deduplicating massive visually identical jpeg image data. *Big Data Res*. 2021;23(1):1–9. <https://doi.org/10.1016/j.bdr.2020.100171>.
41. Fang W, Sheng VS, Wen X, Pan W. Meteorological data analysis using mapreduce. *Sci World J*. 2014. <https://doi.org/10.1155/2014/646497>.
42. Xue SJ, Pan WB, Fang W. A novel approach in improving i/o performance of small meteorological files on hdf5. *Appl Mech Mater*. 2012;117:1759–65. <https://doi.org/10.4028/www.scientific.net/AMM.117-119.1759>.
43. Li T, Wang L, Ren Y, Li X, Xia J, An R. An efficient method for meteorological nephogram recognition in cloud environment. *EURASIP J Wirel Commun Netw*. 2019;2019(1):1–10. <https://doi.org/10.1186/s13638-019-1611-1>.
44. Emanuel KA, Raymond DJ. The representation of cumulus convection in numerical models. 1st ed. Boston: Springer; 1993. p. 02108.
45. Coelho CAS. Comparative skill assessment of consensus and physically based tercile probability seasonal precipitation forecasts for brazil. *Meteorol Appl*. 2013;20(2):236–45. <https://doi.org/10.1002/met.1407>.
46. Pezzi LP, Cavalcanti IFA, Mendonça AM. A sensitivity study using two different convection schemes over south America. *Rev Bras Meteorol*. 2008;23(2):170–89.
47. Lim Y, Schubert SD, Reale O, Lee M, Molod AM, Suarez MJ. Sensitivity of tropical cyclones to parameterized convection in the nasa geos-5 model. *J Clim*. 2015;28(2):551–73. <https://doi.org/10.1175/JCLI-D-14-00104.1>.

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)